

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO**

Helmuth Grossmann Júnior

**UM SISTEMA ESPECIALISTA PARA AUXÍLIO AO
DIAGNÓSTICO DE PROBLEMAS EM
COMPUTADORES UTILIZANDO
RACIOCÍNIO BASEADO EM CASOS**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Mauro Roisenberg, Dr.

Orientador

Florianópolis, dezembro de 2002.

UM SISTEMA ESPECIALISTA PARA AUXÍLIO AO DIAGNÓSTICO DE PROBLEMAS EM COMPUTADORES UTILIZANDO RACIOCÍNIO BASEADO EM CASOS

Helmuth Grossmann Júnior

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Conhecimento, aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Mauro Roisenberg, Dr.
Orientador

Prof. Fernando A. Ostuni Gauthier, Dr.
Coordenador

Banca Examinadora

Prof. Jorge Muniz Barreto, Dr.
Presidente da Banca

Prof. Luiz Fernando Jacinto Maia, Dr.
Membro da Banca

Prof. João Bosco da Mota Alves, Dr.
Membro da Banca

Agradecimentos

Gostaria de agradecer ao prof. Mauro por trazer sempre a luz e a direção certa nos momentos decisivos do desenvolvimento do trabalho.

À minha esposa Fabiane, que soube compreender e aceitar as necessidades de tempo e dedicação que um mestrado impõe e me apoiou incondicionalmente.

Ao meu filho Eduardo, que nasceu durante a realização desta empreitada e com certeza foi o meu maior incentivo para seguir sempre em frente não importando as dificuldades surgidas.

Aos meus pais, que sempre souberam o valor do saber e me apresentaram uma das grandes verdades da vida, que tudo pode ser retirado de um homem, menos o seu conhecimento e a sua fé.

Aos amigos Fabrício e Charles, que me orientaram e ajudaram a desvendar alguns mistérios do Delphi durante o desenvolvimento do protótipo.

Sumário

Lista de Abreviaturas	viii
Lista de Figuras.....	ix
Lista de Tabelas	x
Resumo	xi
Abstract	xii
1. Introdução	13
1.1. Motivação	13
1.2. Objetivos	14
1.3. Organização do Texto.....	14
2. Sistemas Especialistas.....	16
2.1. Definições de sistemas especialistas e suas características	16
2.2. Histórico do SE	17
2.3. Aplicações dos SE	17
2.4. Classificação dos Sistemas Especialistas.....	18
2.5. Arquitetura de um Sistema Especialista.....	21
2.5.1. Base de Conhecimento.....	21
2.5.2. Máquina de Inferência	22
2.5.3. Sistema de Consulta.....	24
2.5.4. Mecanismo de Aprendizagem.....	25
2.5.5. Sistema de Justificativa	25
2.5.6. Quadro -negro	26
2.6. Shell´s para Sistemas Especialistas	27
2.7. Implementações Existentes.....	28
3. Raciocínio Baseado em Casos	30

3.1. Definição de Raciocínio Baseado em Casos.....	30
3.2. Histórico do RBC.....	31
3.3. Diferenças do RBR e do RBC.....	32
3.4. Arquitetura de um Sistema RBC	33
3.4.1. A Base de Conhecimento	33
3.4.1.1. Representação de um Caso	33
3.4.1.2. Indexação.....	36
3.4.2. Recuperação	38
3.4.2.1. Algoritmos de Recuperação	38
3.4.2.2. Casamento e Classificação	39
3.4.3. Reutilização	42
3.4.4. Revisão.....	44
3.4.4.1. Avaliação da Solução Proposta.....	44
3.4.4.2. Reparação da Solução	45
3.4.5. Armazenamento	45
3.5. Vantagens do paradigma RBC sobre os demais	46
3.6. Áreas de Aplicação do RBC e Implementações Existentes.....	47
 4. Raciocínio Baseado em Casos Aplicado a um Sistema de Auxílio ao Diagnóstico	
de Problemas em Computadores.....	50
4.1. Motivação	50
4.2. Aquisição do Conhecimento: o Banco de Dados Disponível.....	50
4.3. Apresentação do Banco de Dados	51
4.3.1. O Cabeçalho e o Rodapé da Ordem de Serviço	53
4.3.2. A Seção <i>Serviço a ser prestado / Problema que o equipamento apresenta</i>	53
4.3.3. A Seção <i>Equipamentos e acessórios recebidos</i>	53
4.3.4. A Seção <i>Descrição dos serviços prestados</i>	54
4.3.5. A Seção <i>Pecas e componentes substituídos</i>	55
4.4. A Filtragem dos Problemas Reais	55
4.5. O Ordenamento dos Casos.....	58
4.5.1. Os Equipamentos Mais Frequentes	58
4.5.2. Os Problemas Mais Frequentes.....	62
4.6. Montando a Base de Casos.....	63

4.7. Apresentação da Proposta de Algoritmo de Casamento	64
4.8. Considerações sobre os Problemas do Domínio.....	66
5. Modelagem e Implementação do Sistema SADPC	67
5.1. Uma Visão Geral das Técnicas Adotadas	67
5.2. Apresentação do Sistema SADPC	68
5.3. Os Usuários do Sistema	69
5.3.1. Os Especialistas	70
5.3.2. Os Engenheiros do Conhecimento	70
5.3.3. Os Usuários Leigos	70
5.4. A Representação de um Equipamento: a Arvore H/S.....	70
5.4.1. A Árvore Hardware/Software: Organização e Indexação.....	71
5.4.2. Cadastrando um novo equipamento	71
5.4.3. Escolhendo um equipamento existente	72
5.4.4. Quando não Sabemos o Equipamento Envolvido.....	73
5.5. A Representação dos Casos	73
5.5.1. Base de Casos: Organização e Indexação.....	74
5.5.2. Cadastrando um Novo Caso	74
5.6. A Iteração de um Novo Caso com o Sistema	75
5.6.1. A Apresentação das Possíveis Soluções.....	77
5.6.2. Quando a Solução Proposta é Válida	78
5.6.3. Quando a Solução Proposta não é Válida	79
5.7. Relatórios do Sistema	79
5.7.1. Listagem das Soluções Possíveis em um Novo Caso.....	79
5.7.2. Relação de casos ainda sem solução	80
5.8. Avaliando o Protótipo	81
6. Conclusões e Considerações Finais	84
6.1. Conclusões.....	84
6.2. Recomendações e Considerações Finais	84
Anexo 1. A Árvore de Equipamentos do Sistema	86
Anexo 2. O Arquivo de PDD's do Sistema	87

Anexo 3. O Arquivo de Casos do Sistema.....	91
Anexo 4. O Código-Fonte do Algoritmo de Recuperação e Casamento – parte 1	98
Anexo 5. O Código-Fonte do Algoritmo de Recuperação e Casamento – parte 2	101
Anexo 6. O Código-Fonte do Algoritmo de Recuperação e Casamento – parte 3	104
Bibliografia	107

Lista de Abreviaturas

BD	Base de Dados
CBR	Case-Based Reasoning
COIN	Corporate Information Network
CPU	Central Processing Unit
H/S	Hardware/Software
IA	Inteligência Artificial
IAS	Inteligência Artificial Simbólica
ICAS	Intelligent Case- Authoring Support
OS	Ordens de Serviço
PDD	Palavras Descritivas de Defeitos
RBC	Raciocínio Baseado em Casos
RBR	Raciocínio Baseado em Regras
RNA	Rede Neural Artificial
SE	Sistemas Especialistas
SOHO	Small Office – Home Office
TI	Trabalho Individual
UFSC	Universidade Federal de Santa Catarina
WWW	World Wide Web

Lista de Figuras

Figura 2.1. A estrutura de um sistema especialista.....	20
Figura 2.2. Estrutura e atualização de uma base de conhecimento	21
Figura 2.3. Exemplo de máquina de inferência	22
Figura 2.4. Interação do engenheiro do conhecimento	24
Figura 2.5. O papel do quadro negro no sistema especialista.....	26
Figura 3.1. As divisões do processo do RBC	30
Figura 3.2. O Ciclo RBC	32
Figura 3.3. O Ciclo RBC – versão mais analítica	33
Figura 3.4. O Processo de Recuperação na Base de Casos	38
Figura 4.1. Leiaute da Ordem de Serviço	50
Figura 4.2. Fluxograma do algoritmo de casamento proposto	63
Figura 5.1. A tela de acesso ao sistema	66
Figura 5.2. A tela principal do sistema	66
Figura 5.3. A tela de manutenção da tabela de usuários.....	67
Figura 5.4. A tela de inclusão de novos equipamentos.....	70
Figura 5.5. A tela do programa de cadastro de novos casos	73
Figura 5.6. Selecionando o equipamento com defeito.....	74
Figura 5.7. Selecionando Palavras Descritivas de Defeitos	74
Figura 5.8. Apresentação das possíveis soluções encontradas	75
Figura 5.9. O fluxograma do algoritmo de casamento	76
Figura 5.10. Listagem das soluções possíveis em um novo caso	78
Figura 5.11. Listagem dos casos ainda sem solução	79

Lista de Tabelas

Tabela 2.1. Diferenças fundamentais entre um sistema de apoio à decisão (SAD) e um sistema especialista (SE).....	18
Tabela 4.1. Palavras que indicam equipamentos com problemas	54
Tabela 4.2. Divisão do banco de dados de ordens após a filtragem.....	54
Tabela 4.3. Comparação dos métodos de filtragem manual e automático	55
Tabela 4.4. Composição do banco de ordens após a segunda filtragem.....	55
Tabela 4.5. Palavras com maior ocorrência na seção <i>Equipamentos e acessórios recebidos</i>	57
Tabela 4.6. Equipamentos e acessórios das ordens de serviço após filtragem.....	58
Tabela 4.7. A primeira versão da árvore HS	59
Tabela 4.8. Palavras com maior ocorrência na seção <i>Problema que o equipamento apresenta</i>	60
Tabela 4.9. Amostra das combinações da palavra não	61
Tabela 4.10. Divisão das OS por equipamento.....	62
Tabela 5.1 A estrutura do arquivo de usuários	67
Tabela 5.2. A estrutura do arquivo de equipamentos	69
Tabela 5.3. A estrutura do arquivo de casos	72
Tabela 5.4. Resultado da avaliação do protótipo SADPC	80

Resumo

Hoje vivemos numa sociedade dependente de computadores, e isso faz com que grandes transtornos sejam causados toda vez que surge um problema no funcionamento dos equipamentos de informática, sendo desejável seu conserto rápida e precisamente.

Para resolver problemas em computadores e periféricos temos hoje em dia técnicos especializados, que levantam as possíveis causas para o problema a partir de seus sintomas e com alguns testes e eliminação de possibilidades chegam à solução. Quanto mais experiência acumulada estes técnicos têm, mais rapidamente eles chegam a solução e com maior grau de acerto. Esta abordagem leva a alguns problemas, como o tempo e dinheiro necessários para formar um bom técnico e o risco que se corre de perder seu conhecimento acumulado quando ele se aposenta, morre ou troca de profissão. Assim, uma solução apropriada para consolidar a memória histórica dos diagnósticos de problemas de computadores é o desenvolvimento de um sistema especialista que utilize o conhecimento armazenado nos sistemas de registro de problemas para propor soluções para um problema corrente.

Este trabalho apresenta um sistema que utiliza o paradigma de raciocínio baseado em casos, desenvolvido com o uso de um banco de dados de 10000 ordens de serviço de problemas reais resolvidos. Os problemas típicos deste domínio, a abordagem adotada e os resultados obtidos com o protótipo construído são descritos.

Abstract

Today we lived in a dependent society of computers, and as result any fault that occurs on these machines raises great troubles to their users, requiring a quick and accurate repair.

To solve faults in computers and peripheral devices nowadays, we have specialized technicians that raise the possible causes for the fault by analyzing its symptoms to then with some tests and elimination of possibilities come up with the solution. The more experience these technicians have the quicker they come up with the solution and with a higher accurateness. This approach brings some problems with it, such as the time and money necessary to form good technicians and the risk of lose the accumulated knowledge when they retire, die or change career. This way an appropriate solution to consolidate historical memory of the computer problems' diagnoses is the development of a specialist system that uses the knowledge in the registries systems to propose solutions to a current problem.

This work that uses the reasoning paradigm based on cases, developed with the use of a database of 10000 orders of service of solved real problems. The typical problems of this domain, the used approach and the results obtained with the built prototype are described.

1. Introdução

1.1. Motivação

Na nossa sociedade atual o computador já pode ser encarado como ferramenta indispensável de trabalho. Inúmeras são suas virtudes e aplicações, aumentando a produtividade daqueles que o usam e fazendo com que fiquemos, de certa forma, dependentes de sua ajuda. Por um lado isso é muito bom, haja visto por exemplo o crescimento assombroso da economia americana na última década, apoiada, segundo a maioria dos especialistas em macro-economia, na informatização maciça dos processos produtivos das empresas que trouxeram ganhos de produtividade jamais vistos. Por outro lado, quando um computador pára por algum defeito, seu usuário com frequência não consegue produzir mais nada enquanto o problema não é resolvido. Os problemas ocorridos nos computadores podem ser originados por inúmeras causas, e a heterogeneidade dos equipamentos e das tecnologias envolvidas num sistema computacional levam a uma complexidade crescente na solução de seus problemas.

Tradicionalmente quando um problema aparecia o usuário do computador chamava um técnico, que usando seus conhecimentos e sua experiência localizava o defeito e o corrigia. No caso de empresas de maior porte, os usuários costumam contar com um departamento especializado no atendimento ao usuário de informática, normalmente chamado de Help Desk, que é acionado quando necessário. Empresas menores ou usuários domésticos contam com empresas terceirizadas ou profissionais autônomos que fazem esse mesmo papel.

Em virtude do aumento da complexidade e da variedade dos problemas, o conhecimento acumulado necessário para que um técnico resolva de forma satisfatória os defeitos que surgem ficou muito grande, e esse conhecimento fica obsoleto com muita rapidez. Sente-se a necessidade de uma forma de armazenar a experiência dos técnicos de uma maneira automatizada, e que essa experiência possa ser resgatada quando necessária, de uma forma inteligente, buscando dentre todo conhecimento guardado aquele que servirá para solução da situação apresentada. Assim os técnicos teriam um sistema computacional de apoio na busca de soluções de problemas ocorridos em computadores.

A partir da apresentação desse contexto, fica clara a necessidade da busca de uma ferramenta que trabalhe com um grande volume de dados, guardados sob a forma de experiências, e que tenha a capacidade de separar com critérios semelhantes aos que um humano usaria, as experiências necessárias para se lidar com cada nova situação que surja. Esse trabalho apresenta a proposta de um sistema que faça o papel desta ferramenta de suporte ao trabalho dos técnicos de informática e dos usuários de computadores com bom conhecimento.

Quando se fala em computadores usando critérios humanos ou agindo com comportamento inteligente semelhante ao humano fala-se em Inteligência Artificial (IA), a área da Ciência da Computação que estuda maneiras de atribuir às máquinas comportamento inteligente. De acordo com as necessidades expostas anteriormente, a

abordagem de IA feita pelos Sistemas Especialistas (SE) é a mais adequada, já que permite a um sistema computacional agir como se fosse um especialista de uma área específica, no caso desejado um técnico especialista em computadores

A partir do exposto ficam claras as motivações que levaram ao desenvolvimento do presente trabalho, que poderiam ser resumidas assim:

- normalmente o conhecimento necessário para resolver um problema atual se encontra em experiências anteriores assemelhadas;
- a formação de um bom técnico leva muito tempo e dinheiro, e um Sistema Especialista agilizaria o processo;
- o técnico experiente tem seu conhecimento somente no seu cérebro e pode morrer ou trocar de emprego levando seu saber consigo;
- o novo conhecimento inserido na base está disponível instantaneamente para todos usuários, disseminando soluções velozmente.

1.2. Objetivos

O trabalho aqui desenvolvido tem por objetivo geral estudar a aplicabilidade dos Sistemas Especialistas que utilizam o Raciocínio Baseado em Casos como ferramenta de apoio ao diagnóstico e ao encontro de soluções para problemas surgidos em microcomputadores.

Como objetivos específicos buscamos:

- a disseminação do estudo e do uso do RBC;
- avaliar o atual estado da arte do RBC;
- propor novas funcionalidades em SE's que usam RBC específicas para o domínio do conhecimento estudado; e
- o desenvolvimento de um protótipo funcional de um sistema computacional que demonstre o RBC em ação.

1.3. Organização do Texto

O texto do presente trabalho foi dividido em 6 capítulos visando uma melhor organização e também para facilitar a exposição das idéias nele contidas. O primeiro capítulo consiste nesta introdução.

O capítulo 2, que dá seqüência à introdução, traz o estudo dos SE em profundidade, mostrando como os mesmos podem ser usados para a construção de um sistema como o desejado. Também será mostrado que os Sistemas Especialistas podem ser construídos a partir de vários paradigmas de raciocínio, sendo o mais adotado o Raciocínio Baseado em Regras (RBR).

No sistema proposto pelo presente trabalho, o paradigma de raciocínio escolhido não foi o RBR, mas sim o do Raciocínio Baseado em Casos (RBC), que está explicitado no capítulo 3, onde também são expostas as razões para esta escolha e as principais diferenças entre o RBR e o RBC.

O capítulo 4 descreve a base de dados disponível para a construção do SE almejado e o trabalho desenvolvido encima da mesma visando à adequação, classificação e filtragem das informações relevantes que virão a alimentar o banco de casos do protótipo modelado e implementado no capítulo 5.

Apresentamos então no capítulo 5 a análise e implementação do modelo proposto de sistema especialista resultado do presente trabalho, desde sua modelagem e desenvolvimento da codificação até os resultados práticos extraídos do mesmo. São apresentadas algumas telas do protótipo visando esclarecer como este funciona e como ele se comporta com exemplos práticos.

O sexto e último capítulo traz as conclusões extraídas da análise comparativa do desempenho do sistema frente ao trabalho de especialistas humanos avaliando então se é viável um sistema desta natureza como ferramenta de apoio ao técnico de informática envolvido no conserto de um defeito em um computador ou periférico. Também são sugeridos caminhos para trabalhos futuros visando complementar e entender este aqui apresentado.

2. Sistemas Especialistas

Este capítulo enfoca os assuntos referentes à revisão bibliográfica sobre sistemas especialistas e busca mostrar qual é o atual estado da arte dos SE, apresentando seus conceitos, características, estrutura e aplicações, servindo como referencial para análise da sua aplicabilidade no presente trabalho.

2.1. Definições de sistemas especialistas e suas características

Várias são as definições encontradas nos livros para Sistemas Especialistas, dentre as quais podemos citar SCHILDT (1987) que diz que “SE são programas de computador que procuram imitar o comportamento humano em alguma especialidade. Para isto eles utilizam informações fornecidas pelo usuário para formar uma opinião sobre um determinado assunto. Desta forma o sistema especialista faz perguntas e o usuário responde, até que ele possa identificar uma hipótese que se adapte às respostas”.

Conforme o FERREIRA (2001), a palavra sistema, dentre 22 sentidos diferentes possíveis, na informática seria um “conjunto de programas destinados a realizar funções específicas.” Já a palavra especialista, de 5 sentidos possíveis 4 se encaixam em nossa área de interesse, significando “1. Pessoa que se consagra com particular interesse e cuidado a certo estudo. 2. Pessoa que se dedica a um ramo de sua profissão. 3. Pessoa que tem habilidade ou prática especial em determinada coisa. 4. Conhecedor, perito.” Podemos perceber claramente que um sistema especialista será um conjunto de programas que realizarão a função de uma pessoa conhecedora profunda de um determinado ramo do saber.

O professor Edward Feigenbaum apud TEIXEIRA (2000) da Universidade de Stanford, um dos principais pesquisadores em sistemas especialistas, definiu um Sistema Especialista (SE) como:

“...um programa inteligente de computador que usa conhecimento e procedimentos inferenciais, para resolver problemas que são bastante difíceis de forma a requererem, para sua solução, muita perícia humana. O conhecimento necessário para atuar nesse nível, mais os procedimentos inferenciais empregados, pode considerar-se um modelo da perícia aos melhores profissionais do ramo. O conhecimento de um sistema especialista consiste em fatos e heurísticas. Os fatos constituem um corpo de informação que é largamente compartilhado, publicamente disponível e geralmente aceito pelos especialistas em um campo. As heurísticas são em sua maioria privadas, regras pouco discutidas de bom discernimento (regras do raciocínio plausível, regras de boa conjectura), que caracterizam a tomada de decisão em nível de especialista na área. O nível de desempenho de um sistema especialista é função principalmente do tamanho e da qualidade do banco de conhecimento que possui”.

Já segundo PARTRIDGE e HUSSAIN (1995) um SE é um programa de computador que utiliza o conhecimento de especialistas humanos capturados em uma base de conhecimentos para solucionar problemas que normalmente necessitam de especialização humana.

WEISS e KULIKOWSKI (1988) afirmam que um SE é aquele que lida com problemas complexos do mundo real, que requeiram a interpretação de um especialista,

e soluciona estes problemas através do uso de um modelo computacional racional de um especialista humano, chegando às mesmas conclusões que este especialista humano chegaria se enfrentasse um problema comparável.

2.2. Histórico do SE

Já na década de 1950 os pesquisadores das Ciências da Computação haviam começado a estudar formas de se desenvolver a área da Inteligência Artificial, trabalhando, segundo CARVALHO (1990) apud MÜLLER (1998), com a lógica matemática e a teoria das funções recursivas, guiando a formulação de processamento de listas e da própria linguagem LISP, que fornece um interpretador para desenvolver expressões simbólicas recursivas que capacitaram o surgimento de sistemas práticos de computação simbólica. Surgiram também nessa época computadores interativos que tornavam possíveis ambientes computacionais para o desenvolvimento e depuração de programas cognitivos. Em outra frente de pesquisa, nesta mesma época, os psicólogos cognitivos criaram um caminho padrão no processo de investigação do raciocínio, modelando o aparente processo de tomada de decisão em termos de regras de produção condicionais. Segundo BARRETO (1999), na década de 1950 “...a admiração com as inúmeras possibilidades destas máquinas fizeram com que elas fossem também conhecidas como cérebros eletrônicos. Isto criou um certo mito que fez com que os computadores fossem olhados com uma certa desconfiança”.

Para destruir o mito criado e popularizar os computadores, no início dos anos 60 uma intensa propaganda de fabricantes conseguiu fazer com que os computadores passassem a ser considerados como incapazes de qualquer forma de raciocínio. Isso ajudou a mostrar que apesar de não terem inteligência, os computadores poderiam ser úteis em tarefas correntes como contabilidade, administração de empresas, preparação de folhas de pagamento, etc. Os computadores passaram a ser cada vez mais vendidos e usados, como máquinas capazes de automatizar tarefas diárias, sempre guardando sua burrice como grande qualidade. Porém, alguns grupos de pesquisadores, nos laboratórios, continuaram a busca da possibilidade de prover computadores de raciocínio inteligente. A partir do anúncio do projeto japonês dos computadores de 5ª geração passou-se a falar em uma nova família de computadores inteligentes rodando aplicações conhecidas como sistemas especialistas. Data dos anos 60 o famoso sistema especialista MYCIN, capaz de diagnosticar doenças infecciosas do sangue em pacientes com a mesma precisão da média dos médicos especialistas da área.

2.3. Aplicações dos SE

Os sistemas especialistas podem ser aplicados com sucesso nas mais diversas áreas do domínio do conhecimento humano, como na medicina, engenharia, ciências administrativas e outras. Segundo RIO e FERNÁNDEZ (1990), existem alguns pontos dessas áreas onde existe uma grande expectativa de se receber altos benefícios com a adoção de sistemas especialistas, tais como:

- Incremento de produtividade, já que o usuário pode chegar mais rapidamente a uma solução para seus problemas com o auxílio do SE e também contar com uma segunda opinião imediata;

- Redução de custos de treinamento, pois é muito mais rápido ensinar uma pessoa a usar um sistema especialista que tenha uma interface intuitiva, ergonomicamente eficiente e amigável do que ensinar tudo que consta da base de conhecimento da aplicação. Isso é particularmente válido para domínios de conhecimento complexos.

- Redução de custos de pessoal. Esse ponto pode ter uma conotação cruel se pensarmos que com o uso de SE podemos trocar especialistas experientes e com altos salários por novatos inexperientes, que ganham pouco e aprendem a usar o sistema. Por outro lado podemos pensar que uma população carente, com poucos recursos, pode contar por exemplo com um sistema de apoio ao diagnóstico médico manipulado por profissionais de baixo custo que resolvam grande parte dos problemas apresentados.

- Aplicação de conhecimento constante na empresa através dos tempos. O SE serve como um grande depósito da experiência acumulada ao longo do tempo numa organização, e este armazém pode ficar a disposição dos usuários, de forma organizada;

- Preservação do conhecimento desenvolvido pelos especialistas da empresa ou organização. Como foi dito no item anterior, a base de conhecimentos de um sistema especialista é um grande armazém contendo todo conhecimento desenvolvido por especialistas e sistematizado por engenheiros de conhecimento ou sistema automatizado equivalente;

- Maior garantia da aplicação da experiência e do conhecimento. Um sistema especialista evita que usuários amadores tentem resolver um problema com conhecimento empírico, não dotado de nenhum embasamento científico. Com a possibilidade de contar com ajuda especializada, a tendência é que os usuários leigos façam uso da mesma.

- Apoio na solução de problemas e possibilidade de uma segunda opinião. Muitas vezes um profissional deve tomar uma decisão acerca de um problema sem ter possibilidade de consultar outras pessoas, pelos mais diversos motivos como: escassez de tempo, assunto sigiloso, inexistência de outros profissionais, etc. Nesses casos, poder contar com uma opinião racional, que sintetiza de forma científica um domínio de conhecimento, é uma grande ajuda na busca da melhor solução para um problema.

2.4. Classificação dos Sistemas Especialistas

Segundo WIDMAN (1998), os SE podem ser classificados de dois modos diferentes:

- **Apoio à decisão:** o programa ajuda o tomador de decisões experiente a lembrar-se de diversos tópicos ou opções, que se considera que ele saiba, mas que possa ter esquecido ou ignorado. Este é o uso mais comum em medicina.

• **Tomada de decisão:** toma a decisão no lugar de uma pessoa, pois isso implicaria algo que está acima de seu nível de treinamento e experiência. Este é o uso mais comum em muitos sistemas industriais e financeiros.

Essa classificação não deve permitir que se confunda Sistemas Especialistas com Sistemas de Suporte a Decisão. Vários aspectos e características atribuídas aos sistemas especialistas podem se equiparar aos atributos de sistemas de suporte a decisão, mas há um limite em que um aplicativo deixa de ser voltado ao suporte à decisão e passa a ser considerado um sistema especialista. Ambos os tipos de sistemas são projetados para a assistência ao usuário e a tabela a seguir, retirada de YANG (1995) pode ajudar a resumir as diferenças fundamentais e evitar confusões.

Característica	Sist. Apoio a Decisão	Sistemas Especialistas
1. Origem do sistema.	Disciplina das ciências da administração e da decisão.	Disciplina da Inteligência Artificial.
2. Objetivo do sistema.	Suporte à intuição de quem toma decisão.	Replicar um assistente humano e substituí-lo.
3. Características da área problema.	Áreas amplas e mal-especificadas.	Áreas de domínio restrito e problemas bem definidos.
4. Tipo de problema tratado.	Situações momentâneas e únicas.	Assistência em áreas de problemas repetitivas.
5. Representação do processo de solução do problema.	Representação esparsa do processo de tomada de decisão.	Representação densa do processo de tomada de decisão.
6. Controle e interatividade (possibilidade de intervenção do usuário).	Usuário confronta um problema. Iniciativa é sempre do usuário.	Modelo do processo de solução do problema; a iniciativa é sempre do sistema.
7. Usuários do sistema.	Desde o topo executivo até os gerentes de linha.	Cliente não especialista, estudante, interessados em SE, especialista que quer um assistente.
8. Tipos de conhecimentos nas bases de dados e de conhecimento.	Fatos.	Fatos, conhecimentos de procedimentos, heurística e outros (julgamentos, conhecimentos causais, meta-conhecimento).
9. Capacidade de raciocínio.	Não possui capacidade de raciocínio.	Exibe alguma capacidade de raciocínio.
10. Capacidade de explanação.	O usuário confia na qualidade da informação.	O usuário executa um papel passivo no processo de resolução do problema e alcance da solução.
11. Fonte de conhecimento.	Fontes reconstruídas (estatísticas, ciência da administração e modelos quantitativos.)	Fontes autênticas (as escolhas de especialistas) e fontes reconstruídas.
12. Natureza do suporte.	Suportes individual, de grupo e organizacional.	Assistência e explanação para um indivíduo (ou um grupo)

Tabela 2.1. Diferenças fundamentais entre um sistema de apoio à decisão (SAD) e um sistema especialista (SE).

Já conforme RIBEIRO (1987), os SE podem ser classificados, quanto às características de seu funcionamento em:

- **Interpretação:** são sistemas que inferem descrições de situações a partir da observação de fatos, analisando dados e procurando determinar relações e seus significados.

- **Diagnósticos:** são sistemas que detectam falhas oriundas da interpretação de dados. A análise dessas falhas pode conduzir a uma conclusão diferente da simples interpretação de dados. Estes sistemas já têm embutidos o sistema de interpretação de dados. Devem permitir ao diagnosticante a decisão sobre medidas a tomar.

- **Monitoramento:** este sistema deve interpretar as observações de sinais sobre o comportamento monitorado. Deve verificar, de maneira contínua, um determinado comportamento em limites preestabelecidos, sinalizando quando forem requeridas intervenções para o sucesso da execução.

- **Predição:** a partir de uma modelagem de dados do passado e do presente, este sistema permite uma determinada previsão do futuro. Como ele baseia sua solução na análise do comportamento dos dados recebidos no passado, deve ter mecanismos para verificar os vários futuros possíveis, a partir da análise do comportamento desses dados, usando raciocínios hipotéticos e verificando a tendência de acordo com a variação dos dados de entrada.

- **Planejamento:** o sistema prepara um programa de iniciativas a serem tomadas para se atingir um determinado objetivo. Possui características parecidas com o sistema para a predição e normalmente opera em grandes problemas de solução complexa. O princípio de funcionamento, em alguns casos, é por tentativas de soluções, cabendo a análise mais profunda ao especialista que trabalha com esse sistema.

- **Projeto:** tem características parecidas com as do planejamento, e deve-se confeccionar especificações tais que sejam atendidos objetivos dos requisitos particulares. Deve ser capaz de justificar a alternativa tomada para o projeto final, e de fazer uso dessa justificativa para alternativas futuras.

- **Depuração:** trata-se de sistema que possui mecanismos para fornecerem soluções para o mau funcionamento provocado por distorções de dados. Faz verificação automática nas diversas partes, com mecanismos para ir validando cada etapa necessária em um processo qualquer.

- **Reparo:** normalmente trabalha em conjunto com um sistema de diagnóstico, desenvolvendo e executando planos para administrar os reparos. São poucos os sistemas desenvolvidos, pois consertar alguma coisa do mundo real é uma tarefa muito complexa.

- **Instrução:** tem mecanismos para verificar e corrigir o comportamento do aprendizado de estudantes. Seu funcionamento consiste em ir interagindo com o treinando, em alguns casos apresentando uma pequena explicação e, a partir daí, ir sugerindo situações para serem analisadas pelo treinando. Pode-se ir aumentando a complexidade das situações e ir encaminhando o assunto, de maneira didática.

- **Controle:** é um sistema que governa o comportamento geral de outros sistemas (não apenas de computação). É o mais completo, de um modo geral, pois deve

interpretar os fatos de uma situação atual, verificando dados passados e fazendo uma predição do futuro. Apresenta os diagnósticos de possíveis problemas, formulando um plano ótimo para sua correção, que será executado e monitorado para que o objetivo planejado seja alcançado.

2.5. Arquitetura de um Sistema Especialista

A seguir serão apresentados os módulos constituintes de um sistema especialista ideal e completo, segundo a maioria dos autores consultados. Deve-se ter em mente que os SE implementados no mundo real podem não ter todos estes módulos ou apresentar algum complemento não mostrado aqui, sendo a arquitetura presente só uma referência.

Ainda destaca-se o fato que esta estrutura aqui mostrada é típica dos SE's que utilizam raciocínio baseado em regras, mudando bastante no caso de SE's que utilizam raciocínio baseado em casos, como veremos no próximo capítulo.

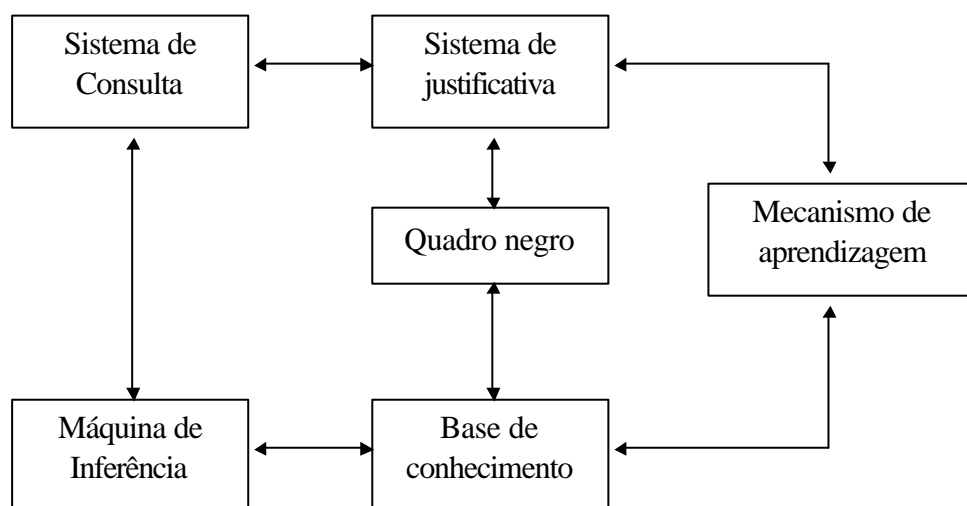


Fig. 2.1. Arquitetura de um Sistema Especialista.

2.5.1. Base de Conhecimento

A base de conhecimento de um Sistema Especialista é, sem sombra de dúvida, o seu componente mais importante e o coração do sistema. Um SE terá tanto conhecimento quanto for armazenado em sua base, e ele só será capaz de tratar uma situação que seja coberta, diretamente ou por inferência, pelo conhecimento armazenado. Conforme RIBEIRO (1987), em um SE com raciocínio baseado em regras, a base de conhecimento será composta por fatos e regras, como podemos ver na figura 2.2.

Na figura vemos dois tipos de usuários que utilizam o sistema especialista. O primeiro é quem utiliza o conhecimento armazenado. Ele fornece fatos, através de uma interface colocada entre esse usuário e o sistema de modo a tornar natural o seu uso e que coloca esses fatos de maneira que o sistema os possa usar em seu funcionamento. O segundo tipo de usuário é o especialista, ou seja, o profissional que detém o conhecimento e o fornecerá sob a forma de regras (nesse exemplo da figura) para que o sistema o use. O conhecimento do especialista é guardado sob uma forma padrão definida por quem projetou e construiu o sistema, de modo que o conhecimento possa ser tratado de maneira mecânica. Este conhecimento normalmente é passado para o sistema através de um editor construído especialmente para ser usado pelo especialista na transferência do seu conhecimento para o sistema e o formata de maneira adequada.

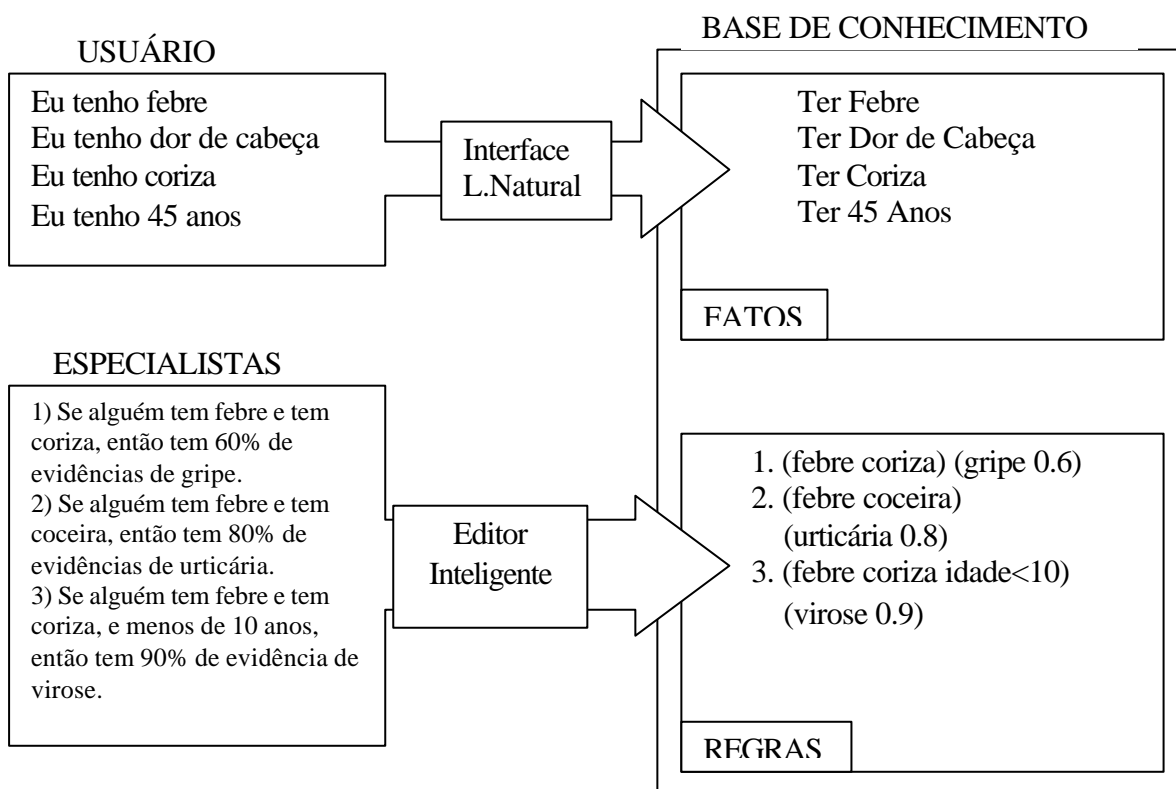


Figura 2.2. Estrutura e atualização de uma base de conhecimento.

2.5.2. Máquina de Inferência

Juntamente com a base de conhecimento, a máquina de inferência compõe o núcleo do sistema especialista. Basicamente se trata de um protocolo de programa para navegar através das regras e dados da representação do conhecimento buscando resolver o problema. Sua tarefa é selecionar e aplicar a regra mais apropriada em cada passo de execução do sistema, o que contrasta com a programação convencional, onde o programador seleciona a ordem na qual o programa deverá executar os passos ainda em tempo de programação.

O papel da máquina de inferência é semelhante ao do especialista humano: quando este recebe uma solicitação de pesquisa ou através de um diálogo para troca de informações, a máquina tenta interpretar a necessidade do usuário, e ajudá-lo a definir melhor aquilo que está procurando e chegar a uma conclusão. Segundo RIBEIRO (1987), existem diversos processos para se fazer a inferência, pois este está diretamente ligado à representação do conhecimento, porém, o método mais usado é o da avaliação de regras. A figura 2.3. ilustra o tipo de busca que uma máquina de inferência pode fazer numa base de conhecimento com representação em árvore. O sistema do exemplo busca identificar o tipo de objeto sobre uma mesa a partir da identificação de três variáveis: som, cor e forma. O exemplo foi retirado do material de aula do Prof. Dr. Paulo BORGES (2001).

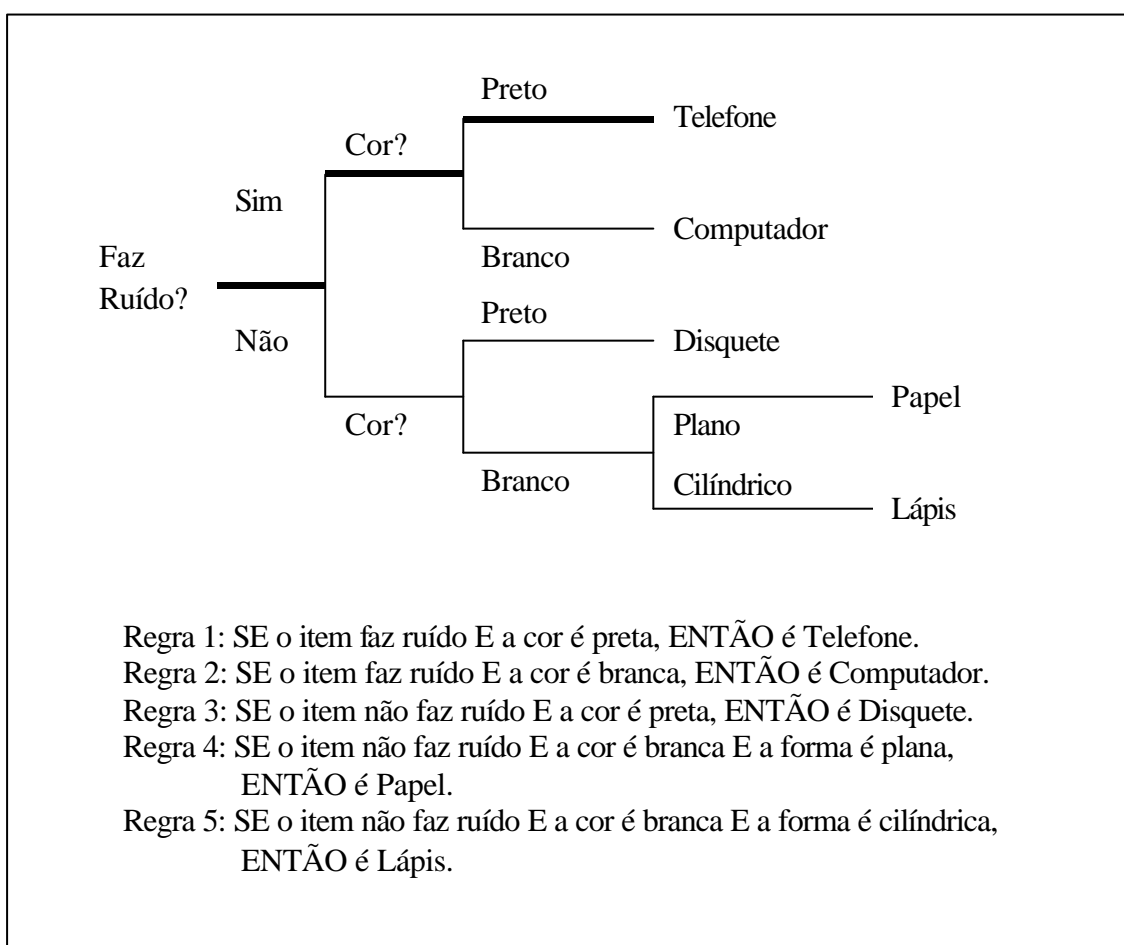


Figura 2.3. Exemplo de máquina de inferência.

Há sistemas que operam através das regras de inferência caminhando para frente na ordem das mesmas, outras caminham para trás e outros empregam ambas as direções.

Isabela MÜLLER (1998), citando RAMOS (1995) define que o encadeamento para frente, também conhecido como raciocínio dirigido a eventos, é utilizado quando a

solução de um problema tem dados ou idéias básicas como ponto de partida, sem uma meta previamente definida. Por exemplo quando temos um conjunto de sintomas e queremos inferir as possíveis doenças associadas.

Já o encadeamento para trás, também chamado de raciocínio dirigido a metas, é utilizado quando se tem uma meta como ponto de partida e se deseja satisfazê-la. A partir da meta todas as hipóteses são verificadas. Por exemplo, tendo um determinado efeito, investiga-se os fatores que em tese levariam a confirmar este efeito. Um problema que pode acontecer neste método é a explosão combinatória a que se pode chegar em alguns casos, quando seria o usuário confrontado com uma extensa apresentação de possibilidades para sua análise, o que não seria muito viável.

Outra possibilidade é a combinação dos dois métodos. Esta abordagem é usada quando se tem amplo espaço de busca e esta busca pode ser dividida hierarquicamente, sendo então possível combinar o uso das direções “do inferior para o maior” e “do maior para o inferior” apropriadamente.

No próximo capítulo, quando será analisado o raciocínio baseado em casos veremos que a busca realizada pela máquina de inferência sobre uma base de casos, ao invés de regras e fatos, utilizará outro método, que buscará qual é o caso dentre os armazenados que mais se aproxima do problema apresentado pelo usuário através da interface do sistema.

2.5.3. Sistema de Consulta

Este módulo também pode ser chamado de interface com o usuário, e é acionado basicamente em três situações: no início da interação, quando o usuário vai informar ao sistema qual é o problema para o qual ele busca uma solução; durante o processamento, quando o motor de inferência necessitar alguma orientação ou intervenção direta do usuário indicando um caminho a seguir; e ao final do processamento, apresentando o resultado obtido para o utilizador.

Nesta interface do usuário, um componente que pode ser muito importante para o sucesso do sistema é a capacidade do mesmo compreender sentenças escritas em linguagem natural. Este sub-módulo do sistema de consulta tem o objetivo, segundo RIBEIRO (1987), de tornar transparente para quem usa o sistema toda sua complexidade. Seu objetivo é uma comunicação o mais perto possível da linguagem normalmente usada entre dois interlocutores humanos, evitando que o usuário do sistema tenha que aprender uma linguagem computacional própria ou complexa para o uso do sistema. Este módulo é, ele próprio, um sistema especialista, e é um estudo à parte em Inteligência Artificial.

2.5.4. Mecanismo de Aprendizagem

O conhecimento de um especialista é adquirido durante sua formação e vida profissional de diversas formas, leituras, experiências e casos similares. Este conhecimento está armazenado permanentemente pelos especialistas em seu cérebro e tem diversas origens, tal como: análise estatística de dados (heurística), tentativa e erro (experiências), leituras, palestras, trocas de experiências com outras pessoas.

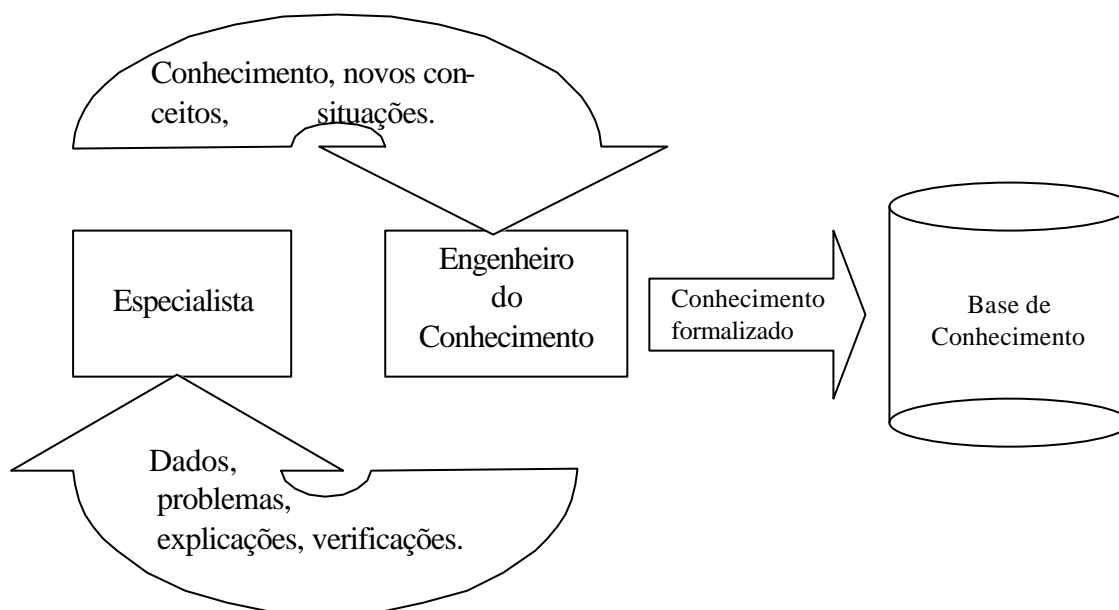


Figura 2.4. Interação do engenheiro do conhecimento.

Em última análise, o que se verifica é que o aprendizado vem do processo de experiência, e de seus resultados experimentais. É portanto de fundamental importância que os sistemas especialistas sejam capazes de aprender e fazer crescer o seu conhecimento a partir de um conhecimento básico sobre o assunto. Segundo RIBEIRO (1987), o processo ideal de aprendizado seria aquele que o conhecimento do especialista fosse absorvido diretamente pelo sistema, mas o que se verifica na prática é a presença de uma figura intermediária, conhecida como engenheiro de conhecimento, que prepara o conhecimento para ser armazenado em uma forma própria, e também fornece as explicações necessárias dos conceitos utilizados, como podemos ver na figura 2.4.

2.5.5. Sistema de Justificativa

O sistema de justificativa visa explicar ao usuário do SE como se chegou à conclusão apresentada. Pode também ser capaz de explicar porque está fazendo determinada pergunta ao usuário se for questionado a respeito. O modelo de justificação

busca apresentar ao usuário as informações necessárias para que este fique totalmente a par de como e por que uma estratégia de pesquisa foi aplicada e uma decisão foi alcançada.

Conforme RIBEIRO (1987), este módulo utiliza diversos recursos e estruturas próprias para atender ao seu objetivo, mostrando que regras e que fatos foram usados da base de conhecimento na busca da solução. As implementações atuais do justificador procuram atender às perguntas: como? Por quê? Por que não? Qual?

Segundo o mesmo autor não existe ainda, entre os pesquisadores, uma uniformidade de conceitos básicos. Alguns dos problemas mostrados abaixo ainda são bastante discutidos e são implementados de maneira muito diferente e rudimentar, nos sistemas existentes; ou de maneira mais radical, simplesmente não são implementados. As naturezas dos problemas são:

- Quem irá receber a explicação?
- A que nível esta explicação deve ser dada?
- Que detalhes deverão ser mostrados?
- Como será mostrado
- Qual o universo de explicações que o sistema poderá vir a dar?

2.5.6. Quadro-negro

O quadro negro é o módulo do SE onde é armazenado todo o processo do desenrolar da pesquisa, reformulações e dados parciais obtidos durante o diálogo do sistema, por sua máquina de inferência, com o usuário, através do sistema de consulta.

Segundo RIBEIRO (1987) o quadro-negro é uma área de memória aonde o sistema vai gravando e apagando os dados que vai usando no processo de inferência até chegar a uma solução. Para se chegar a uma solução, há necessidade de se avaliarem regras que são recuperadas da base de conhecimento para uma área de trabalho na memória, nesse local essas regras são ordenadas periodicamente em uma nova ordem para serem avaliadas. Durante essa avaliação deve-se verificar fatos e hipóteses e também há necessidade de uma área onde são guardados os valores das variáveis para se trabalhar tais fatos e hipóteses. As conclusões dessas regras irão gerar novos fatos e novas hipóteses que precisam ser guardadas (gravadas) temporariamente durante o processo de inferência (dedução), em algum local. Essa área de memória usada para execução das operações descritas acima se chama quadro-negro ou rascunho e seu funcionamento aparece resumido na figura 2.5. Resumindo, o quadro-negro, ou rascunho, é uma área de trabalho que pode contar a história de como se desenvolveu uma pesquisa.

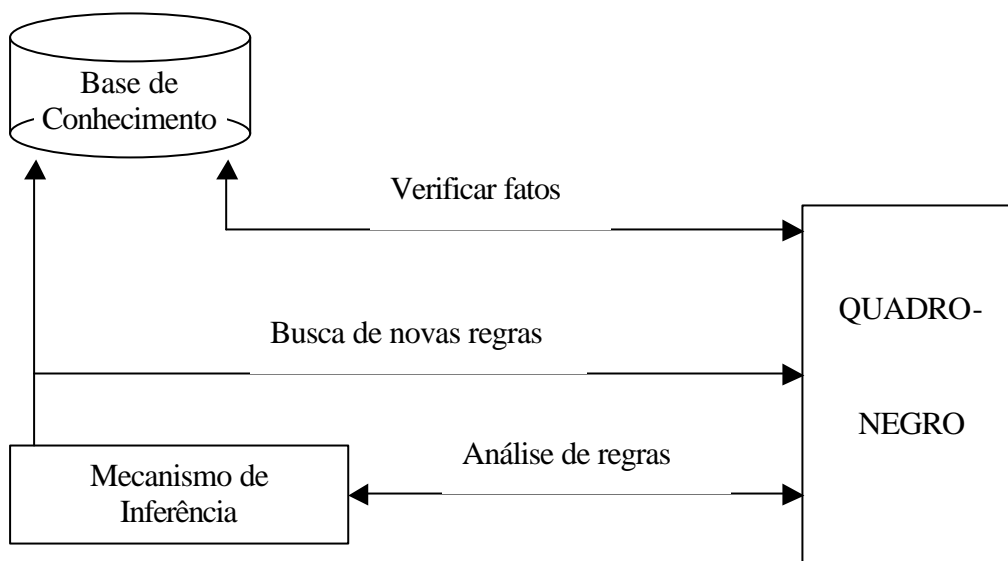


Figura 2.5. O papel do Quadro-Negro no SE.

2.6. Shell's para Sistemas Especialistas

Conforme CHORAFAS (1988), um programa shell pode ser encarado como se fosse uma caixa de ferramentas de componentes que formam um SE. Dessa forma, se quisermos criar um sistema especialista em pouco tempo e com um esforço relativamente menor, basta escolhermos um shell adequado às nossas necessidades e preenchê-lo com uma base de conhecimentos, fatos e regras, que representem o domínio de conhecimento tratado. Esta escolha do shell a ser usado normalmente é condicionada por restrições: a área de aplicação na qual recai o domínio do SE, a possibilidade e facilidade na expressão de fatos e relações pertinentes ao problema, as facilidades de interface da linguagem e a forma de raciocínio usada para buscar a solução na base de conhecimentos.

Com o uso de um shell, o desenvolvimento de um protótipo de SE é bastante acelerado e contamos também com as seguintes vantagens, dentre outras:

- ajuda a clarear as idéias do construtor do SE;
- é valioso para a realimentação do usuário;
- pode ser útil no projeto de núcleos de testes num estágio inicial.

Mas os shells também têm suas desvantagens. Uma camada de interface é adicionada ao sistema, que requer seu próprio interpretador residente e resulta em lentidão no processamento. Também se deve lembrar que os shells são ferramentas de uso genérico e provavelmente conterão módulos desnecessários no seu SE específico e outros módulos poderão faltar. Outro detalhe que pode ocorrer é que diferentes projetos de SE numa mesma entidade podem escolher shells diferentes, resultando numa proliferação de diferentes linguagens de programação que dificulta o desenvolvimento à manutenção dos sistemas.

Após a exposição desses fatos, conclui-se que os shells são ótimas ferramentas para o desenvolvimento inicial dos SE, mas se estes vão ser usados intensamente e por um longo tempo, vale a pena otimizar o protótipo inicial, reescrevendo o sistema de uma forma mais enxuta e funcional.

2.7. Implementações Existentes

Incontáveis são as aplicações baseadas no paradigma dos sistemas especialistas hoje em dia, principalmente as que usam raciocínio baseado em regras, e algumas das mais famosas e representativas são brevemente aqui descritas.

- **MYCIN:** segundo BARRETO (1999), é um SE da autoria de E. H. Shortliffe para diagnóstico e tratamento de doenças infecciosas do sangue. O MYCIN se tornou famoso por ser pioneiro no uso de um conjunto de conceitos de serviram de base para a moderna estrutura dos SE. Alguns desses conceitos são: separação da base de conhecimento e da máquina de inferência, o que permite usar um mesmo motor de inferência com uma base de conhecimentos diferente e tratar um domínio de conhecimento totalmente diverso do original; tratamento da incerteza através de fatores de certeza, que mostraram que o raciocínio médico não segue um padrão probabilístico; foi o primeiro de uma série de SE da área médica.

- **PROSPECTOR:** é um SE de apoio para geólogos em busca de depósitos com recursos minerais. Seguiu muitas das idéias apresentadas pelo MYCIN, introduzindo uma interface com o usuário que permitia a comunicação em linguagem técnica de geologia, chamada Lifer. Um ponto inovador do PROSPECTOR foi a divisão dos fatos e regras da base de conhecimento em 15 modelos, cada um com cerca de 200 fatos e regras específicos de cada tipo de depósito mineral.

- **ACE:** automated cable expertise, é um sistema especialista desenvolvido pela AT&T que contém conhecimento destilado em forma de regras *se-então*. Este conhecimento foi obtido de peritos na manutenção de cabos das companhias telefônicas. Diferencia-se de outros SE por dois pontos: manipular grandes quantidades de dados e obter esses dados automaticamente de bases de dados distribuídas.

- **ABEL:** sistema especialista de aconselhamento terapêutico para identificação de distúrbios eletrolíticos e ácidos-básicos.

- **Art of Negotiating:** simula uma sessão de negociação e orienta na modelagem de uma estratégia, baseando-se em perguntas sobre a personalidade dos participantes da negociação.

- **Communication Edge:** análise e síntese para preparação de reuniões de negócios.

- **Emes:** administração de distribuição de força elétrica de componentes elétricos em naves espaciais.

- **NTC:** consultor para eliminação de problemas de rede capaz de dar análises interpretativas de problemas da Ethernet/Decnet.

- **SCCES:** cálculo de riscos de fatores envolvidos no processamento do petróleo e desgaste e corrosão.
- **VM:** interpretação e consulta sobre terapia intensiva em relação a pacientes em estado grave.
- **XCON:** o Expert Configurer é um SE utilizado diariamente nas operações mundiais de manufatura da Digital para configurar a maioria dos computadores VAX e PDP-11 que estão em pedido. Tem um módulo que permite que os vendedores configurem interativamente sistemas computadorizados e tem também um módulo para treinar o pessoal de vendas.

3. Raciocínio Baseado em Casos

Este capítulo faz uma explanação sobre o que é o raciocínio baseado em casos segundo alguns autores da área, seu histórico de desenvolvimento, estrutura de funcionamento e quais são as principais diferenças entre o raciocínio baseado em casos e o raciocínio baseado em regras. Ao final do capítulo são apresentados alguns sistemas especialistas já implementados que usam esta forma de raciocínio.

3.1. Definição de Raciocínio Baseado em Casos

Conforme SCHANK (1982) “O Raciocínio Baseado em Casos é aquele utilizado por sistemas artificiais inteligentes que raciocinam recuperando situações ocorridas no passado quando se encontrar frente a novos problemas”. Um estudante resolve problemas de matemática, por exemplo, relembrando problemas anteriormente estudados com características semelhantes e tentando aplicar o método que usou para resolver o problema de antes, alterando-o para a nova situação.

Um caso representa na terminologia RBC, segundo AAMODT (1994) uma situação ocorrida anteriormente — denominada caso anterior ou do passado — que é capturada e aprendida de modo a ser utilizada na solução de problemas futuros — denominados casos correntes ou novos. Os casos, que podem ser representados em diferentes formatos e tamanhos, reproduzem o conhecimento específico para um determinando contexto, registrando o conhecimento a nível operacional. De acordo com KOLODNER (1993), um caso registra uma experiência que tem o potencial de ser útil para atingir um objetivo ou conjunto de objetivos mais facilmente no futuro ou de alertar para a possibilidade de falhas imprevistas em um problema.

Segundo o Dicionário Aurélio, FERREIRA (2001), a palavra raciocínio vem do latim *ratiociniu* e pode significar “1. Ato ou efeito de raciocinar. 2. Encadeamento, aparentemente lógico, de juízos ou pensamentos. 3. Capacidade de raciocinar; juízo, razão; racionalidade. 4. Processo discursivo pelo qual se passa de proposições conhecidas ou assumidas (as premissas) a outra proposição (a conclusão) à qual são atribuídos graus diversos de assentimento.”

Já a palavra caso pode assumir 8 significados diferentes segundo o mesmo dicionário, sendo que duas se encaixam nos propósitos da ciência da computação e do RBC (mais especificamente ainda a primeira) que seriam “1. Acontecimento, fato, sucesso, ocorrência. 2. Eventualidade, conjuntura, hipótese.”

Podemos derivar destas definições que o raciocínio baseado em casos é o encadeamento lógico de pensamentos racionais que se baseiam em acontecimentos transcorridos.

Segundo FOX (1995), o processo do RBC pode ser dividido em tarefas separadas, porém onde exatamente ficam as linhas que separam cada tarefa depende de cada aplicação em particular. Em linhas gerais normalmente esse processo se divide em três partes: a recuperação de um caso semelhante da base de casos, a avaliação do caso

recuperado frente às necessidades do novo caso, e a adaptação do caso recuperado para que atenda satisfatoriamente o novo problema, conforme podemos ver na figura 3.1.

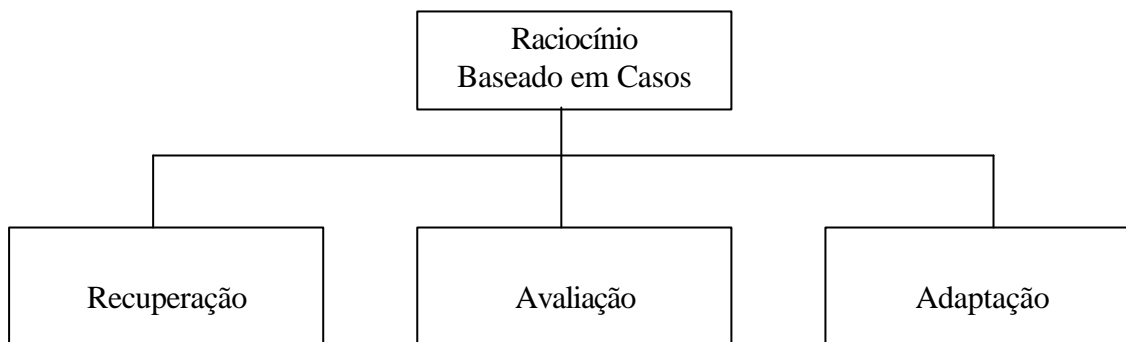


Figura 3.1. As divisões do processo do RBC.

3.2. Histórico

Em 1977, conforme MELCHORS (1999), Schank e Abelson propuseram que o conhecimento geral das pessoas sobre as situações está armazenado em scripts, permitindo que elas criem expectativas sobre o que ouvem, e dessa forma façam deduções sobre as relações entre as coisas que ouviram. Os scripts foram propostos como uma estrutura de memória conceitual, descrevendo informação sobre eventos estereotipados, como ir a um restaurante ou a uma consulta a um médico. Porém, experimentos com scripts mostraram que eles não representam uma teoria completa de representação de memória, já que as pessoas confundem eventos que tem scripts similares. Os scripts parecem ser apenas uma faceta dos conhecimentos que as pessoas utilizam para a compreensão: elas se valem também outros tipos de conhecimento, como o conhecimento sobre objetivos, planos, relações interpessoais e papéis efetuados pelas pessoas, segundo KOLODNER (1993). Representações sobre estes tipos de conhecimento têm sido propostas e sistemas de computadores que usam estes tipos de conhecimento para o entendimento têm sido desenvolvidos.

Em 1982, Roger SCHANK (1982), propôs a teoria da Dinamic Memory (memória dinâmica) e o papel central que a lembrança de situações anteriores – casos – e modelos de situações (scripts, MOPs — pacotes de organização de memória) desempenham na resolução de problemas. Este trabalho, segundo AAMODT e PLAZA (1994), apresenta as raízes do uso de raciocínio baseado em casos na Inteligência Artificial. Outras características do campo RBC têm sido originadas do estudo de raciocínio analógico proposto por Gentner, ainda conforme MELCHORS (1999) e das teorias de formação de conceitos, resolução de problemas e aprendizado experimental dentro da filosofia e psicologia.

O primeiro sistema RBC, denominado CYRUS, foi desenvolvido por KOLODNER (1993). Este sistema, baseado no modelo de memória dinâmica de

SCHANK (1982) e na teoria MOP de resolução de problemas e aprendizado, foi basicamente um sistema de questões e respostas com conhecimento de várias viagens e encontros do primeiro secretário do estado americano, Cyrus Vance. O modelo de memória de casos desenvolvido por este sistema serviu de base para outros diversos sistemas RBC, dentre eles MEDIATOR, PERSUADER, CHEF, JULIA e CASEY.

Outra base de RBC e conjunto de modelos foi desenvolvida por Bruce Porter e seu grupo na University of Texas, em Austin. Inicialmente trabalhando com o problema de aprendizado automático para classificação de tarefas, o grupo desenvolveu o sistema PROTOS. Esse sistema enfatiza a integração do conhecimento geral do domínio e do conhecimento específico de casos em uma estrutura de representação unificada — um modelo de memória de casos. Além dessa, outra contribuição importante para a área foi o trabalho do grupo de Edwina Rissland na University of Massachusetts, em Amherst, que desenvolveram o sistema HYPO, aplicado para o domínio do Direito.

Atualmente, as pesquisas em RBC têm estendido-se rapidamente, sendo percebido um crescente número de artigos envolvendo RBC em periódicos de IA e um aumento de aplicações comerciais de sucesso, além de pesquisas em desenvolvimento em diversos países.

3.3. Diferenças do RBR e do RBC

As principais diferenças entre as duas metodologias são:

- enquanto as regras são pequenas partes do conhecimento, independentes e consistentes, casos são grandes porções do conhecimento, provavelmente com partes redundantes entre si.
- para determinar a solução de um problema em um sistema RBR, as regras precisam casar exatamente entre si e há necessidade da aplicação de várias regras para chegar à solução. Em RBC é exigido apenas um casamento parcial entre os casos e a solução é proposta como um todo para depois ser adaptada.
- a manutenção de um sistema baseado em regras é necessária em todo o seu ciclo de vida, pois a adição de uma regra frequentemente requer a modificação de várias outras. A adição de casos em RBC (aprendizado) requer pouca ou nenhuma modificação, tornando o aprendizado bem mais simples.

O mais forte argumento para a utilização do paradigma baseado em casos para a simulação de um especialista humano é o fato de o que faz um especialista é sua experiência. Se o conhecimento especialista fosse somente aquele dos livros, bastaria modelar a informação dos livros em regras e pronto. Entretanto, num processo de extração de conhecimento especialista, observa-se que o especialista coloca sua experiência para tentar explicar como faz seu trabalho. Muitas vezes, sua especialidade é ilustrada através de exemplos, experiências passadas que justificam a abordagem escolhida para tratar determinado problema. Além disso, quando é possível estabelecer uma regra para determinada situação, esta vem sempre acompanhada de exceções.

3.4. Arquitetura de um Sistema RBC

Podemos representar a arquitetura de um sistema RBC de diversas formas, de maneira mais sintética ou mais analítica, de acordo com a necessidade do estudo em questão. Logo a seguir apresentamos duas figuras representando o processo RBC, a primeira ligeiramente mais sintética que a segunda, mas como podemos acompanhar ambas representam o mesmo processo. Seguindo as figuras temos explicações acerca de cada componente do processo do raciocínio baseado em casos.

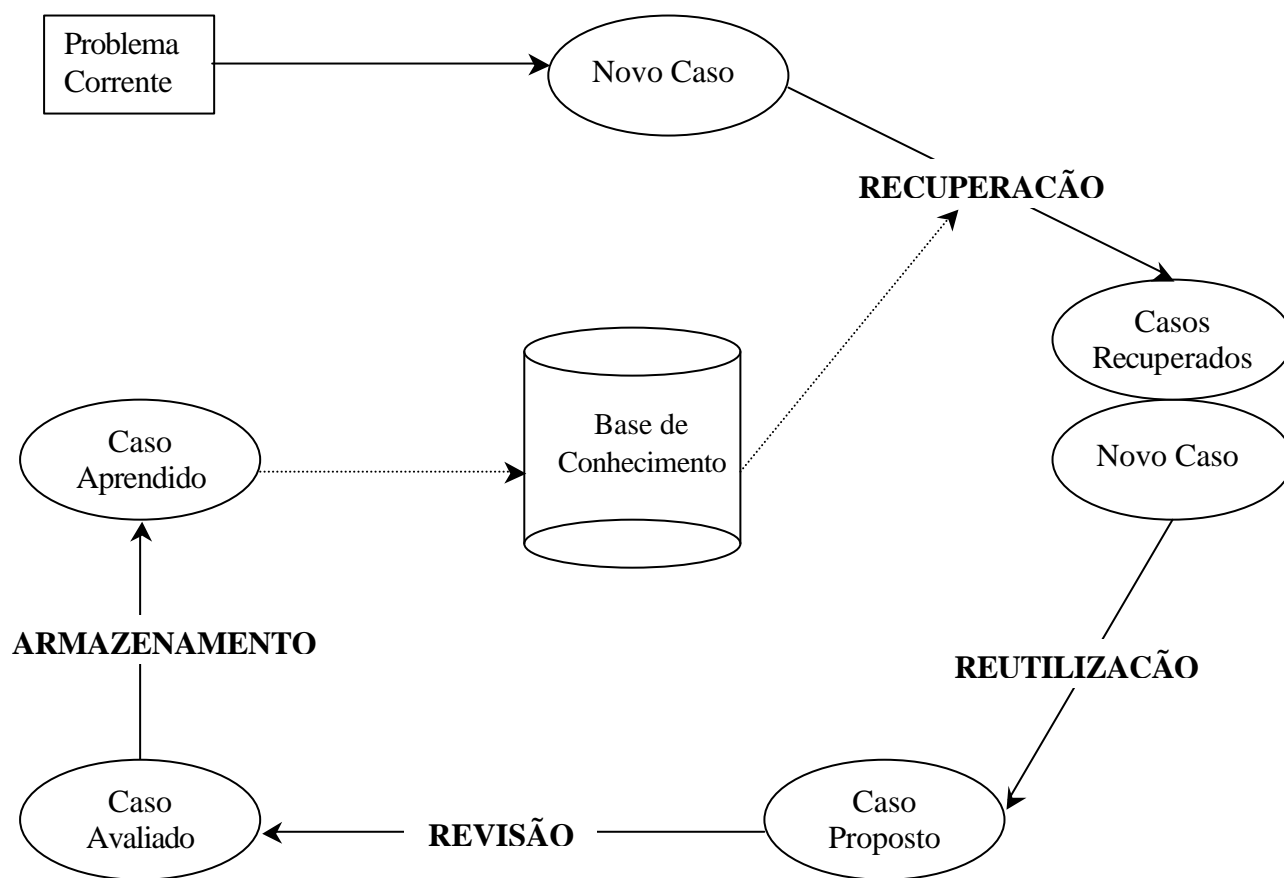


Figura 3.2. O Ciclo RBC

3.4.1. A Base de Conhecimento

3.4.1.1. Representação de um Caso

Os casos representam um conhecimento específico relacionado a um contexto, armazenando conhecimento em nível operacional. Devem ser armazenados, entretanto, apenas os casos que possuem diferenças dos demais, de modo que possam acrescentar algum conhecimento útil para o sistema. Os casos podem apresentar diferentes

tamanhos e modelos, assim como podem representar um pequeno ou grande intervalo de tempo. KOLODNER (1993) define a partir dessas observações que “um caso é um pedaço de conhecimento contextualizado representando uma experiência que ensina uma lição fundamental para atingir os objetivos do raciocinador”.

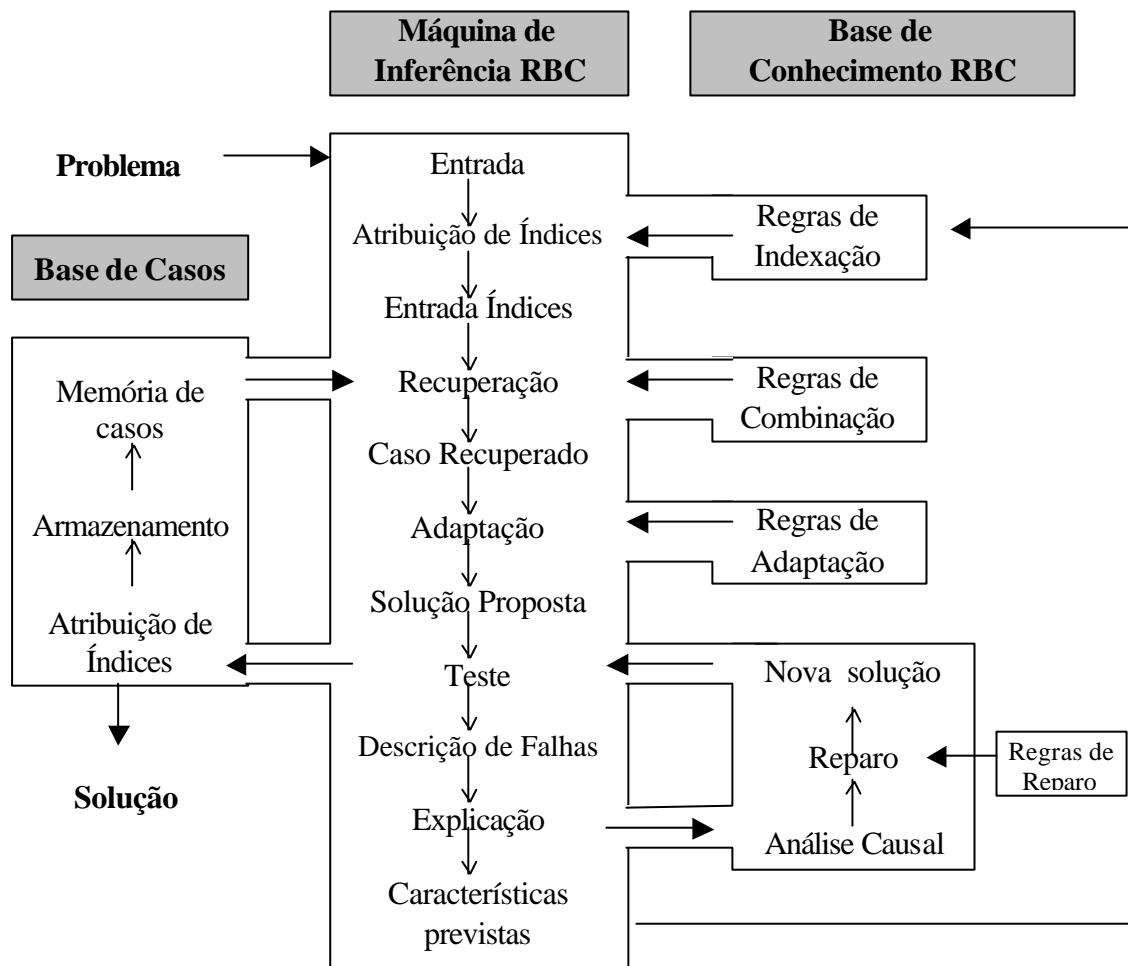


Fig. 3.3. O Ciclo RBC – versão mais analítica.

Os casos são formados por duas partes maiores: as lições que ele ensina — contidas no conteúdo dos casos — e o contexto em que pode ensinar tais lições — representado pelos índices. Tipicamente, o conteúdo de um caso é formado por três partes principais, KOLODNER (1993):

- **Descrição do problema ou situação:** apresentando as condições do ambiente no momento que o caso ocorreu e as restrições associadas;
- **Solução:** que expressa a solução derivada para o problema, podendo ser formada por uma ação, um conjunto de procedimentos, um diagnóstico, uma classificação, um projeto, etc., variando conforme o tipo de problema que o sistema se propõe a resolver;
- **Resultado:** que descreve o estado no ambiente real após o caso ter ocorrido e a solução ter sido, assim, aplicada.

A descrição do problema ou situação representa o problema que precisa ser solucionado ou a situação que precisa ser interpretada, classificada ou entendida. Como, de modo geral, um sistema RBC utiliza as similaridades entre a descrição do caso corrente e do caso armazenado para decidir se um caso armazenado deve ser recuperado, a descrição deve ser suficientemente detalhada para que o sistema seja capaz de julgar a aplicabilidade do caso na nova situação.

A descrição do problema é formada por três componentes maiores, de acordo com KOLODNER (1993): pelos objetivos a serem atingidos na resolução de um problema, pelas restrições relacionadas a estes objetivos e pelas características do problema e relações entre as partes.

Os objetivos na descrição do problema se referem às intenções ou propósitos a serem atingidos na situação. Estas intenções podem ser concretas ou abstratas, dependendo de que lições o caso almeja fornecer. Se o caso deseja diagnosticar um conjunto de sintomas, então o objetivo será “diagnóstico”. Já se o caso pretende criar uma solução para uma questão, seu objetivo será “criação”. As restrições, por sua vez, envolvem as condições relacionadas aos objetivos almejados que precisam ser consideradas no processo, como, por exemplo, não incluir um certo ingrediente na criação de uma receita culinária ou não causar determinada reação em um paciente que possui alergia a um medicamento.

Por fim, as características da situação envolvem todas as outras informações descritivas sobre a situação, que são relevantes para a obtenção dos seus objetivos. Em uma aplicação de diagnóstico médico, por exemplo, sintomas e resultados de exames do paciente são descritores necessários para construir a solução, enquanto outras características podem também ser consideradas para a busca do objetivo. Do mesmo modo, alguns sintomas ou particularidades observados em uma rede de computadores devem necessariamente ser usados para solucionar um problema, enquanto outros podem ser usados para a solução desejada.

Dependendo da aplicação sendo desenvolvida, diferente atenção é exigida para cada um dos componentes da descrição do problema — algumas enfatizam os três componentes da descrição, outras enfatizam somente um ou dois. As situações que exigem uma solução explícita, como um projeto ou planejamento de tarefas, enfatizam, de modo geral, os objetivos e as restrições em suas representações do problema.

Aquelas aplicações que requerem entendimento ou interpretação, por sua vez, tendem a ter um único objetivo simples como diagnóstico ou entendimento e enfatizam os descritores da situação problema. Assim, conforme sugere KOLODNER (1993), há dois princípios gerais que devem ser seguidos na decisão de que informações devem pertencer à descrição do problema:

- incluir toda a informação descritiva que deve ser explicitamente levada em conta na busca do objetivo do caso;
- incluir todos os tipos de informação descritiva que são normalmente usados para descrever os casos desta espécie.

O primeiro princípio permite que o sistema ignore características que são relevantes, mas implícitas, por nunca variarem. O segundo, é necessário para a acessibilidade do caso. Ao menos algumas porções da descrição do caso são usadas em um sistema para a indexação e julgamento da similaridade, sendo melhores para a

recuperação aqueles que são semelhantes nas dimensões que se sabe serem relevantes para a solução do problema. Porém, como existe muita informação que não é conhecida quando um novo caso é descrito, é necessário alternar o modo de determinar os melhores casamentos.

A parte componente solução de um caso representa os conceitos ou objetos que atingiram os objetivos descritos na descrição do problema, levando em consideração as restrições e as características especificadas. Existem diferentes tipos de soluções — em um problema de projeto a solução é o artefato elaborado, enquanto que em um problema de interpretação de uma situação ela é a interpretação apresentada para o caso. Além da solução de um caso recuperado ser usada para derivar uma solução para o caso corrente, ela pode também possuir componentes que acrescentem adaptação e revisão da solução proposta. Entre as partes componentes da solução que podem ser úteis para essas funções estão, além da solução propriamente dita, o conjunto de passos usados no raciocínio para resolver o problema, o conjunto de justificativas para as decisões que foram tomadas, as soluções aceitáveis que não foram escolhidas e suas justificativas, as soluções inaceitáveis que foram eliminadas e suas justificativas e as expectativas frente aos resultados após a solução ser efetuada.

Por fim, a parte componente resultado de um caso identifica o que ocorreu como consequência da solução efetuada e como ela foi realizada. Este componente inclui tanto o feedback do ambiente real como as interpretações para este feedback, de modo que o sistema possa antecipar problemas em potencial e prever resultados de uma solução proposta. Além disso, o resultado pode também incluir a explicação do porquê uma solução falhou ou expectativas frente a uma solução falharam e o que foi feito para reparar estas falhas.

Os casos podem ser representados em uma grande variedade de formas usando praticamente que todos os formalismos de representação da Inteligência Artificial, conforme WATSON (1994), incluindo frames, objetos, predicados, redes semânticas e regras, além de estruturas menos ricas semanticamente, como os modelos de banco de dados comerciais. Segundo ABEL (1986), muitos sistemas usam mais de um dos formalismos acima combinados.

3.4.1.2. Indexação

Os índices são normalmente utilizados quando a base de casos é grande, para facilitar a recuperação dos casos armazenados, sendo os responsáveis por tornar um caso acessível no momento e condição apropriado — isto é, quando ele possuir um potencial para contribuir para a solução do problema corrente. São combinações dos descritores importantes de um caso, isto é, daqueles descritores que distinguem um caso dos outros. O esquema de indexação envolve várias partes. Primeiramente, devem ser designados rótulos para os casos no momento em que estes são armazenados na base, de modo a garantir que eles possam ser recuperados no momento apropriado. Tais rótulos são usados no momento de recuperação para julgar se o caso armazenado deve ser selecionado. Uma segunda questão envolvendo a indexação é a definição da organização dos casos, de modo que a busca através da base de casos seja eficiente e precisa. Relacionada a estas questões está a definição dos algoritmos de recuperação a serem utilizados.

A atribuição de índices aos casos depende da compreensão do conteúdo e das informações que um caso pode fornecer. Ela deve permitir que sejam reconhecidas as similaridades entre a situação corrente e os casos armazenados que podem contribuir para atingir os objetivos do caso corrente. Assim, para a escolha de um bom índice, é necessária uma grande compreensão da situação problema, de forma que essas similaridades possam ser corretamente identificadas. Considere-se, por exemplo, a resolução de um problema ocorrido em uma rede de computadores. Informações como a aplicação utilizada e o usuário que identificou o problema não são importantes quando o problema envolve o acesso físico de um equipamento por problemas em sua placa de rede ou no cabeamento. Por outro lado, se o problema envolver a não autenticação do usuário por uma aplicação, a aplicação utilizada e o usuário são dados importantes.

Como apontado em ABEL (1986), a similaridade que se busca em sistemas de raciocínio baseado em casos é muitas vezes diferente daquelas semelhanças superficiais, obtidas pela comparação de dados descritivos. É um estilo de similaridade mais abstrata, que permite reconhecer em diferentes contextos as soluções que possam ser aplicadas em novos casos.

Através da experiência com a construção de sistemas RBC, a comunidade RBC propõe, conforme KOLODNER (1993), alguns princípios que devem ser seguidos para a escolha de índices apropriados:

- eles devem ser preditivos, isto é, devem prever a utilização da informação presente nos casos para diferentes situações problema;
- devem endereçar os propósitos em que o caso pode ser usado;
- devem ser suficientemente abstratos para permitir que um caso seja útil em uma variedade de diferentes situações;
- devem ser suficientemente concretos para que possam ser facilmente reconhecidos em situações futuras.

Os índices podem ser escolhidos através de métodos manuais, onde a escolha começa com a análise dos casos para a identificação da utilidade que poderia ter o caso, e sob que circunstâncias. Essas informações devem ser então traduzidas para representações que o sistema pode usar, definindo um conjunto de descritores, que são então trabalhados de modo a garantir que os índices sejam aplicáveis em âmbito geral e que possam ser reconhecidos no máximo de situações possíveis.

Além dos métodos manuais, existem, também, métodos de indexação automáticos, que são disponibilizados por muitas das ferramentas de RBC disponíveis no mercado. Entre os métodos automáticos referenciados por KOLODNER (1993), podemos citar o método baseado em uma checklist, o método baseado em diferenças e o método baseado em explicação. Entretanto, apesar do sucesso de muitos métodos automáticos, descritos por WATSON (1994), Janet Kolodner alerta que os índices tendem a ser mais bem escolhidos pelo modo manual, e em aplicações práticas devem ser escolhidos desta forma.

3.4.2. Recuperação

3.4.2.1. Algoritmos de Recuperação

Um algoritmo de recuperação de casos é o responsável por encontrar, a partir da descrição de um problema ou situação, um pequeno conjunto de casos similares ao problema corrente que seja útil para a identificação da sua solução. A busca pelos casos similares não deve considerar, porém, apenas a descoberta de algumas dimensões da descrição do problema similares à situação. Na identificação da similaridade entre os casos, segundo KOLODNER (1993), alguns atributos são mais importantes que outros no julgamento da mesma e esta valoração pode variar de acordo com os objetivos almejados pelo sistema. Assim, a recuperação de casos similares envolve considerar que os casos similares ao problema corrente são aqueles que são similares nas dimensões que auxiliam o sistema a realizar suas tarefas ou atingir os objetivos desejados.

A recuperação dos casos úteis à situação corrente envolve várias etapas, cada uma possuindo diferentes pontos que devem ser analisados. Inicialmente, precisam ser identificadas quais as características ou dimensões do caso corrente que devem ser utilizadas para julgar a similaridade dos casos armazenados. Isso é determinado levando-se em conta os propósitos para os quais os casos estão sendo recuperados e as dimensões que foram relevantes no passado para determinar o resultado do ambiente para as soluções aplicadas. Essas características serão então utilizadas pelos procedimentos de casamento e classificação para identificar quais dos casos armazenados têm o potencial de serem mais úteis.

Como os índices de um caso indicam quais das suas dimensões são mais importantes para julgar a similaridade, os algoritmos de casamento utilizam os índices para auxiliar no processo de identificação de que características devem ser consideradas para a similaridade entre os casos. Esses algoritmos devem, contudo, estar habilitados para distinguir, dentre os vários índices existentes, quais focalizar para um dado momento quando os casos estiverem indexados em muitas formas.

Uma outra questão relevante no processo de recuperação é a dos algoritmos de busca propriamente ditos. Estes algoritmos são os processos que viabilizam encontrar na base aqueles casos que potencialmente podem ter um bom casamento com a situação corrente. A partir dos casos identificados pelos algoritmos de busca empregados, os procedimentos de casamento são aplicados.

Os algoritmos de busca são relacionados diretamente às estruturas utilizadas nas bases de casos para organizá-las. Uma estrutura em lista é, por exemplo, consultada pelo algoritmo de modo diferente de uma estrutura em árvore complexa. Assim, diferentes organizações dos casos na base ocasionam diferentes algoritmos para recuperar seus casos.

Um esquema dos processos envolvidos na recuperação de um caso pode ser visto na figura abaixo, adaptada de KOLODNER (1993). Conforme mostra o nível superior da figura, inicialmente é feita uma avaliação da situação e são identificadas as características que serão usadas para a busca na base de casos, sendo freqüentemente necessário que algumas dessas características sejam elaboradas pelo sistema. Os algoritmos de recuperação, através da descrição do problema e dos índices selecionados para a consulta na base, buscam os casos com potencial de serem similares e utilizam os

mecanismos de casamento, seja para calcular o grau de casamento entre a situação corrente e os casos encontrados, seja para calcular o casamento de uma dimensão individual. Os algoritmos de recuperação retornam, então, uma lista dos casos parcialmente casados, cada qual com no mínimo algum potencial de ser útil para o sistema. Os casos são, por fim, analisados pelos procedimentos de classificação e aqueles que possuem mais potencial de serem úteis são retornados.

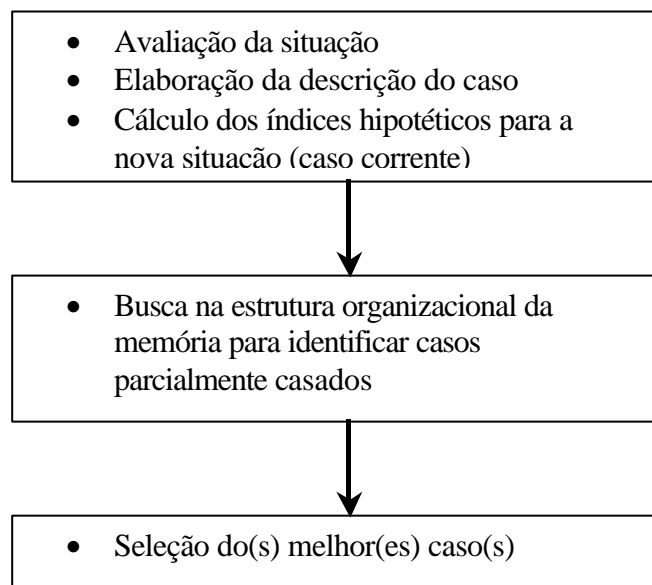


Figura 3.4. O Processo de Recuperação na Base de Casos

3.4.2.2. Casamento e Classificação

Conforme já foi visto, o processo de recuperação dos casos similares da base envolve a definição dos índices a serem utilizados, a busca dos casos com potencial de serem úteis para a situação corrente, o casamento desses casos com o caso corrente e a classificação dos casos recuperados, de modo a selecionar o melhor ou os melhores casos. Esses processos não são, porém, isolados uns dos outros: eles dependem e são algumas vezes altamente integrados entre si.

Muitos dos procedimentos de recuperação para as estruturas discutidas acima são exemplos disso, agregando funções de casamento na própria busca. Algoritmos para busca em memória linear, por exemplo, fazem o casamento de cada caso com o caso corrente, computam o valor de casamento e escolhem o caso de maior valor. Já os algoritmos de busca em redes de características compartilhadas agregam funções de casamento no momento de busca, utilizando relações que fazem o casamento parcial das características do nodo. Os procedimentos de casamento de casos, assim, podem ocorrer

não apenas após a busca na base de casos, mas também serem utilizados para esta busca.

Para a definição dos procedimentos de casamento é necessário que algumas questões sejam consideradas. Uma dessas questões envolve a identificação da correspondência entre os campos do caso corrente e os casos armazenados. Isso pode ser feito por meio da associação de campos com nomes iguais, contudo nem sempre esta associação é tão simples: os casos podem possuir número de características diferentes, estar representados em diferentes pontos de vista, variando com os objetivos almejados pela situação descrita, etc. Nessas situações, o uso de heurísticas de senso comum pode contribuir para a identificação da correspondência. Em KOLODNER (1993), cinco heurísticas que podem ser utilizadas nesse processo são apresentadas:

1. campos com mesmo nome devem ser mapeados diretamente, mesmo considerando que uma estrutura possua subdivisões e outra não;
2. preferir o mapeamento entre as características que possuem o mesmo papel funcional;
3. preferir o mapeamento das características que satisfazem as mesmas restrições;
4. preferir o mapeamento das características que possuem os mesmos valores ou valores similares;
5. múltiplas características do problema podem ser mapeadas para uma característica simples mais geral ou vice versa.

Quando um modelo causal para o domínio está disponível, isso pode também ser utilizado para a definição dessa correspondência. O sistema CASEY, aplicado ao domínio da medicina, é um exemplo de como um modelo causal pode ser usado para determinar a correspondência entre características.

Neste modelo, são utilizadas regras de evidência para determinar como dois campos podem ou não ser considerados iguais. É utilizado um modelo que define que características e resultados correspondentes a outras características e outros resultados, que características levam a outras, que valores entre os possíveis para uma dimensão pertencem ao mesmo intervalo e são assim considerados similares.

Outra questão necessária para os procedimentos de casamento é a definição de como será calculado o grau de similaridade entre as características correspondentes.

Entre os métodos disponíveis citados por KOLODNER (1993) estão:

- distância qualitativa e quantitativa. Este método utiliza várias formas, dependendo do tipo de valor da dimensão:
 - (a) dividir em regiões, atribuindo casamento exato para aqueles valores que estão na mesma e os valores uma região, duas regiões, etc. para aqueles que não se encontrarem na mesma. Para resolver problemas gerados por valores muito próximos dos limites dos intervalos, podem ser criadas regiões que se sobrepõem;
 - (b) medir a diferença numérica e atribuir classificação qualitativa;

- (c) fazer a enumeração de valores qualitativos quando não há números e medir como casamento exato, casamento com uma região distante, duas regiões distantes, etc. Pode também ser atribuído um valor numérico para cada valor qualitativo e compará-los;
- (d) comparar os atributos de cada valor utilizando uma estrutura de comparação que aponta as similaridades e diferenças ou também utilizando representações com frames;
- (e) comparar numericamente os valores numéricos, atribuindo um casamento pior para valores mais distantes.

• julgar o grau de similaridade baseado na eficiência da função. Para isso pode ser utilizado um modelo causal. Se dois valores diferentes qualitativamente (como, por exemplo, uma cadeira e uma pedra utilizada para uma pessoa sentar-se) são mostrados correspondentes por um modelo causal, então elas casam. Neste método não é comparado o grau de casamento, apenas se eles casam ou não.

• hierarquia de abstração. Neste método, a similaridade entre dois valores é medida pela utilização da abstração com valor comum a estes valores mais específica, sendo que quanto mais específica é a abstração comum, maior a similaridade.

Por fim, é necessário atribuir valores de importância para as dimensões da representação, de modo a apontar as informações mais relevantes para a similaridade dos casos sendo comparados. Para isso, devem ser atribuídos pesos para cada campo, o que pode ser feito de forma global, para um grande número de casos, ou atribuído de modo mais local, apenas para um pequeno conjunto. Esta importância entre as dimensões pode ser identificada com a ajuda de um especialista ou através de uma avaliação estatística para determinar quais campos resultam em diferentes resultados ou soluções. Pode também ser interessante atribuir diferentes valores de importância para cada objetivo da situação, o que pode ser feito de forma dinâmica.

Uma vez que alguns casos tenham sido recuperados e os procedimentos de casamento tenham sido efetuados sobre eles, estes devem passar por métodos de classificação a fim de sejam escolhidos os casos mais similares à situação corrente. Como já foi dito, muitas vezes esses procedimentos são feitos de modo integrado, em uma única etapa.

Existem vários métodos envolvendo os procedimentos de casamento e classificação que são utilizados no processo de recuperação dos casos. Entre estes apresentamos:

• **algoritmo de vizinhança** (nearest-neighbor matching): este algoritmo é um procedimento numérico em que é utilizada uma função de avaliação numérica para cada característica. Para cada característica do caso corrente é encontrada a característica correspondente no caso recuperado, os valores são comparados, o grau de casamento é calculado e é multiplicado pelo grau de importância da característica;

• **exclusão**: os casos que não possuem características ou relações que os tornam diferentes ou não úteis em relação ao caso atual são excluídos, como no caso de objetivos diferentes, por exemplo. Após este processo, é realizada uma avaliação numérica.

Existem também métodos que levam em conta o contexto da situação, que estão presentes em sistemas que tratam diferentes propósitos ou contextos e, assim, precisam utilizar meios de avaliar seus graus de casamentos parciais também diferentes. Isso pode ser feito utilizando-se:

- **múltiplas associações de importância**, em uma classificação dinâmica. Neste método são utilizados vários conjuntos de critérios de importância, cada um associado com as condições relacionadas às circunstâncias sobre as quais eles deveriam ser usados. Isso pode ser feito através:

- (a) do fracionamento da base de casos sendo relacionado um diferente conjunto de critérios de importância para cada partição;
- (b) da associação de diferentes critérios de importância para cada diferente objetivo de raciocínio;
- (c) da associação de um ou mais conjuntos de critérios de importância para cada caso, cada um associado com uma diferente tarefa para qual o caso pode ser usado.

Quando somente um objetivo de raciocínio será usado na base de casos, mas o que é importante varia com o tipo da nova situação, a biblioteca de casos pode ser seccionada de acordo com o tipo e para cada partição é indicado um conjunto de valores de importância para as dimensões dos casos (caso a). Quando múltiplos objetivos são suportados pela base de casos, entretanto, o particionamento por tipo não é suficiente e os casos em cada partição devem ser avaliados diferentemente dependendo do objetivo do raciocínio para o qual ele deve servir (caso b). Para acomodar diferentes objetivos de raciocínio, diferentes funções de avaliação podem ser associadas para cada diferente objetivo de raciocínio. Por fim, quando critérios para julgar a similaridade variam tanto em relação ao objetivo sendo buscado como em relação ao tipo de caso, valores de importância devem ser assinalados, levando ambos em conta (caso c). Podem ainda existir, porém, casos altamente particulares. Nessa situação, devem ser associados um ou mais conjuntos de critérios de importância para cada caso. Cada conjunto de valores é designado para um objetivo e enfatiza as dimensões do caso que, se casarem bem, indicam que ele pode servir para o objetivo.

- **preferências para implementar um esquema de classificação relativa.** É utilizado para fazer um corte após os casos terem sido selecionados, passando-os por filtros se existem muitos casos após o casamento. Esses filtros podem ser relacionados ao objetivo de raciocínio, ao grau de especificidade e facilidade de adaptação, e à frequência e ocorrência recente.

3.4.3. Reutilização

Após os casos similares terem sido recuperados da base e o melhor caso ter sido selecionado, a solução deste caso deve ser reutilizada, podendo ser adaptada para as necessidades do problema corrente. A adaptação consiste, segundo WATSON (1994), na identificação das diferenças entre o caso recuperado da base e o caso corrente e a análise das partes do caso recuperado que podem ser transferidas para o caso corrente, aplicando fórmulas e regras que consideram estas diferenças e sugerem uma solução.

Uma solução pode ser reutilizada de modo direto, sendo aplicada a solução recuperada sem ser adaptada. Essa técnica, denominada por alguns de adaptação nula ou identificada simplesmente como cópia, é útil em problemas que envolvem um raciocínio complexo, mas uma solução simples, e é utilizada nos sistemas em que as diferenças não são consideradas relevantes, ao passo que as similaridades entre os casos o são. Entretanto, em sistemas que levam essas diferenças em conta, um processo de adaptação é necessário, podendo ser feito através da simples substituição de componentes da situação recuperada por componentes da situação corrente, por meio da transformação da antiga solução para uma solução que seja aplicada ao novo problema e por meio da derivação de uma nova solução, utilizando métodos para derivar a solução ou pedaços da solução do caso armazenado. Entre os métodos ou técnicas que têm sido usados para adaptação podem ser citados:

- **reinstanciação:** usada para instanciar características da antiga solução com características da nova. É utilizada, por exemplo, segundo KOLODNER (1993), pelo sistema CHEF para propor a substituição de um prato por outro na elaboração de um cardápio de uma refeição.

- **adaptação parametrizada:** usada para o ajuste de parâmetros da antiga solução através do uso de heurísticas especializadas que identificam as relações entre as especificações de entrada e as diferentes soluções. Esta técnica é frequentemente usada quando as variáveis relevantes do caso são numéricas, sendo um exemplo de sua aplicação o cálculo da quantidade de ingredientes pela relação do número de pessoas. Uma das dificuldades em aplicar esta técnica é o conhecimento da função que relaciona as variáveis descritivas do problema e a solução.

- **busca local:** fornece um meio de buscar em uma estrutura de conhecimento auxiliar um substituto para um valor ou estrutura do caso recuperado que não é apropriado para a situação corrente. Um exemplo seria a busca em uma rede semântica por frutas que pudessem substituir a fruta laranja em um prato de sobremesa sendo elaborado.

- **consulta a memória:** procura em estruturas de conhecimento auxiliares ou na memória de casos por algo com uma dada descrição.

- **substituição baseada em casos:** utiliza, para a substituição de um parâmetro, casos da base.

- **transformação de senso comum:** utiliza heurísticas de senso comum para inserir, remover ou modificar componentes em uma solução.

- **reparo guiado ao modelo:** efetua transformações na solução do caso anterior, guiadas por um modelo causal. É indicada especialmente em sistemas que envolvem raciocínio sobre equipamentos, seja envolvendo diagnóstico ou projeto.

- **adaptação ou reparo de propósitos especiais:** utiliza heurísticas de propósitos especiais para modificar estruturas ou adaptações específicas ao domínio não fornecidas pelos demais métodos acima. Essas heurísticas são indexadas pelas situações em que são aplicáveis. Podem ser utilizadas para, por exemplo, inserir um passo extra em um procedimento de solução, se determinado componente fizer parte do problema.

- **adaptação por derivação ou procedural:** utiliza o método usado para encontrar a solução do caso recuperado para derivar a solução da situação corrente.

Usando este raciocínio, um problema pode ser resolvido pela mesma sequência de passos que foi previamente utilizada para solucionar o caso recuperado. É importante observar que procedimentos não são soluções do problema, mas instruções que devem guiar para a causa do problema e soluções potenciais. Com essa forma de adaptação, o sistema oferece (ou mesmo executa) os procedimentos que funcionaram em casos passados para chegar à solução do problema corrente. Para utilizar este método, é necessário haver um repositório dos procedimentos e um modo de mapear tipos de problemas para procedimentos que mais regularmente levam a uma solução.

- **adaptação baseada em crítica:** nesta técnica, o usuário observa a solução proposta no caso recuperado e manualmente adapta a solução para o caso corrente. É usada quando o sistema recupera soluções que o usuário sabe que não funcionará, sabendo porque não funcionará e como fazer funcionar. Nessas situações, o usuário deve adaptar o caso por si mesmo, já que o caso sugerido como uma possível solução pode solucionar a situação com uma pequena modificação. Deve ser, entretanto, documentado o modo que o caso foi adaptado para que ele possa ser usado no futuro, conforme ABEL (1986).

3.4.4. Revisão

Após um caso ter sido recuperado da base e sua solução adaptada para o caso corrente, são necessários dois passos de revisão: a avaliação da solução proposta e, se necessária, a reparação da solução.

3.4.4.1. Avaliação da Solução Proposta

A avaliação da solução é resultado da aplicação da solução proposta em um ambiente de avaliação dos resultados ocorridos. O ambiente pode ser representado por um ambiente de simulação apto a gerar os resultados corretos a uma solução. De modo geral, porém, essa etapa é avaliada no ambiente real, para o problema real. Os resultados da execução da solução podem variar desde alguns segundos, quando aplicada, por exemplo, para corrigir automaticamente uma configuração em uma rede, até vários meses, quando aplicada, por exemplo, a um tratamento médico. Durante o período de execução, segundo MELCHORS (1999), um caso pode ser aprendido, sendo já mantido armazenado na base, porém a informação de que o caso não foi ainda avaliado deve ser marcada a este.

Conforme apresentado por LEWIS (1993) apud MELCHORS (1999), existem, de modo geral, três modelos de execução para uma solução:

- **execução manual:** na execução manual, o usuário do sistema é responsável por interpretar a solução proposta e decidir se ela deve ou não ser executada. Segundo ABEL (1986), ocorre na maior parte dos sistemas RBC, em que o sistema somente sugere, com base na sua experiência, no processo de recuperação e no processo de adaptação, uma boa solução para o problema, que é executada pelo usuário.

- **execução não supervisionada ou sem controle humano:** em alguns domínios, quando uma solução é expressa em um programa de computador, um sistema

RBC pode ter a capacidade de executar a solução que ele propõe, realizando-a automaticamente sem intervenção ou controle humano. Quando isso ocorre, é formado um ciclo fechado de solução de problemas sem intervenção humana, em que o problema é submetido ao sistema, um caso similar é recuperado e sua solução é adaptada para a situação corrente, a solução é executada pelo sistema e os resultados são inseridos na base de casos. Esse tipo de execução envolve, contudo, um alto risco, pois delega muita responsabilidade ao sistema.

- **execução supervisionada ou com controle humano:** um modo intermediário é a execução da solução proposta de modo automático com o controle do usuário. Nessa modalidade, o usuário pode permitir ou proibir a execução de uma solução que é sugerida pelo sistema, que a executa automaticamente se for autorizado.

3.4.4.2. Reparação da Solução

Caso a solução proposta não apresentar resultados satisfatórios durante sua execução, simulada ou não, faz-se a reparação da solução, onde uma nova solução deve ser gerada. Uma vez que a solução encontrada for correta, a experiência obtida deve ser aprendida, sendo armazenada para uso futuro. Esses processos (avaliação, reparação e armazenamento) podem ser vistos como obtenção de experiência, ao passo que o processo de recuperação e de adaptação é visto como uma aplicação da experiência armazenada.

Após uma solução ter sido avaliada, se o seu resultado for satisfatório, a experiência que foi obtida durante o processo de resolução do problema corrente deve ser armazenada no sistema. Isso é feito através do processo de aprendizado, que será visto na próxima seção. Se, entretanto, o resultado da avaliação mostrou que a solução proposta não produziu um resultado adequado, o caso deve ser reparado. A reparação do caso envolve a detecção dos erros da solução corrente e a recuperação ou geração da explicação para a ocorrência destes erros. Um exemplo de um sistema em que isso é realizado é o sistema CHEF, desenvolvido por Hammond e citado por AAMODT (1994), que utiliza um conhecimento causal para gerar explicações do porquê certos objetivos almejados no caso corrente não foram atendidos, e armazena as situações gerais que produziram falhas usando uma técnica de aprendizado baseada em explicação. Um segundo passo no processo de reparação utiliza, então, as explicações das falhas para modificar a solução corrente de modo que os erros não mais ocorram. A reparação de falhas pode ser executada diretamente (quando for garantido que está correta) ou pode ser avaliada e reparada novamente se necessário. O esquema do processo de revisão pode ser visto na figura 2.3.

3.4.5. Armazenamento

O armazenamento dos casos é o processo de incorporação ao sistema da experiência obtida durante a resolução do problema. Como foi visto, esse aprendizado do sucesso ou falha da solução proposta é proporcionado pelo resultado da fase de avaliação e possível reparo da solução. O armazenamento envolve a seleção de que informações do caso devem ser retidas, em que formas estas informações devem ser

retidas, como indexar o caso para posterior recuperação em problemas similares futuros e como integrar o novo caso na memória de casos, segundo AAMODT (1994).

Quando um caso foi resolvido pelo uso de um caso recuperado, um novo caso pode ser construído ou o caso recuperado pode ser generalizado para incluir o caso corrente. Já quando o problema foi resolvido por outros métodos, como por consultas ao usuário, um novo caso inteiro deve ser construído. Informações como descritores relevantes do problema e solução para a situação são dados geralmente utilizados para armazenamento do conhecimento, assim como explicações e outras formas de justificativa do porquê a solução utilizada foi proposta para o problema. No sistema CASEY, por exemplo, explicações são retidas nos casos e reutilizadas futuramente na modificação da solução. Informações obtidas da tarefa de reparo da solução podem também ser extraídas e retidas no sistema, seja como casos separados que falharam, seja dentro de casos problema totais.

Por fim, o sistema RBC deve integrar o conhecimento aprendido à base de casos. Isso pode ser feito com a modificação do esquema de indexação dos casos existentes, em que o sistema pode aprender o modo de ajustá-los para que a similaridade possa ser mais bem identificada. Os pesos ou importância dos índices de um caso ou solução em particular podem ser ajustados pelo sucesso ou falha do uso do caso na solução do problema corrente. As características que foram julgadas relevantes na recuperação do caso com sucesso têm então sua importância de associações aumentada, ao passo que as associações das características que levaram a casos não similares serem recuperados são enfraquecidas. Com esse processo, a estrutura de índices tem a função de canalizar e adaptar a memória de casos para seu uso.

O aprendizado pode ocorrer também dentro de um modelo de conhecimento conceitual geral, por exemplo utilizando outros métodos de aprendizado automático ou através de interação com o usuário, conforme AAMODT (1994). Assim, com uma interface apropriada ao usuário, o modelo de conhecimento geral do sistema pode ser incrementado e refinado naturalmente ao longo da resolução de problemas, do mesmo modo como é feito com a memória de casos.

3.5. Vantagens do paradigma RBC sobre os demais

O RBC pode ser usado para várias tarefas de “raciocínio”, como por exemplo: propor soluções para novos problemas, antecipar, evitar e explicar fracassos em soluções propostas, e adequar e/ou reparar soluções propostas.

- **extração do conhecimento:** basta ter um banco de dados. A difícil fase de extração do conhecimento especialista é facilitada nos sistemas de RBC pois a representação do conhecimento pode ser feita com o preenchimento direto dos fatos que descrevem uma experiência. O paradigma proporciona um excelente mecanismo de aprendizagem, que pode ser utilizado para aquisição automática de conhecimento.

- **representação do conhecimento:** A tarefa de representação do conhecimento resume-se em escolher o tipo de estrutura da base de casos. Em aplicações onde a complexidade exigir, é necessária a criação de índices, entretanto muitas aplicações omitem este passo.

- **reutilização do conhecimento:** o conhecimento contido nos casos pode ser utilizado ou combinado e adaptado para gerar novas soluções além das originalmente presentes na memória.

- **aprendizagem:** a atualização do conhecimento é feita automaticamente, na medida que as experiências são utilizadas, assim o sistema pode crescer e incrementar sua robustez e eficiência.

- **consciência:** estes sistemas possuem uma certa consciência com relação a suas próprias limitações. Se o sistema não encontra casos com a devida similaridade com o caso em análise, o sistema não gera solução alguma. Isto evita possíveis problemas gerados por sistemas sob outros paradigmas que podem gerar uma solução aproximada.

- **fácil acesso às soluções:** Primeiro, estes sistemas podem recuperar uma solução rapidamente, ao contrário de outros sistemas que precisam fazer todo um trabalho para buscar uma solução numa vasta base de conhecimento. Outra vantagem refere-se ao fato de que o problema deve ser identificado pelo sistema o suficiente para recuperar uma solução, não é necessário que o sistema entenda perfeitamente as condições e circunstâncias do problema para propor uma solução. Os sistemas de RBC também proporcionam um meio de resolução de um problema quando não houver um algoritmo disponível para avaliação e solução do mesmo.

- **“raciocínio” implícito:** a incerteza implícita nas informações contidas nos casos é utilizada sem a necessidade de um tratamento específico desta incerteza.

3.6. Áreas de Aplicação do RBC e Implementações Existentes

O raciocínio baseado em casos é uma área efervescente da Inteligência Artificial e, como tal, serve de campo para pesquisas e aplicações dos mais diversos ramos do conhecimento humano. Para exemplificar o que se pode fazer com o paradigma RBC, apresenta-se a seguir uma série de casos de áreas de aplicação e sistemas que foram implementados usando RBC, retirados dos trabalhos apresentados nos workshops da 4ª Conferência Internacional em Raciocínio Baseado em Casos, realizada em julho de 2001 em Vancouver, no Canadá. Sendo essa a mais recente conferência internacional exclusiva do assunto podemos traçar um panorama do que se faz hoje no mundo em matéria de RBC.

- **Gerenciamento do conhecimento orientado a processos:** uma das áreas de pesquisa e desenvolvimento de aplicações mais promissoras do RBC é a do gerenciamento de bases de conhecimento visando o reuso de experiências do setor produtivo das empresas. Um trabalho apresentado por JEDLITSCHKA (2001) mostra uma fábrica experimental criada no Instituto Experimental de Engenharia de Software de Fraunhofer, em Kaiserlautern, Alemanha. Lá foi criado o COIN (Corporate Information Network – Rede de Informações Corporativas), com o objetivo de fornecer aos seus usuários informação/conhecimento na hora certa, representados adequadamente, e dentro de um contexto atualizado, seguindo a filosofia “just-in-time”. COIN é usado como um ambiente de projeto real para o desenvolvimento e validação de tecnologias e métodos para experiências de administração orientadas a objetivos

incluindo elucidação, processamento, disseminação, apresentação, manutenção e avaliação do conhecimento.

- **Ferramentas de suporte a autoria de RBC:** um dos maiores problemas dos sistemas RBC é a chamada autoria, a criação de novos casos na base de conhecimento, esteja ela previamente vazia ou não. Um bom SE que use raciocínio baseado em casos deve ser capaz de recomendar a inclusão de um novo caso em uma biblioteca de casos existente a partir da detecção da representatividade e ineditismo de um caso apresentado por um usuário.

McSHERRY (2001) apresenta uma ferramenta de ICAS (Intelligent case-authoring support – Suporte inteligente a autoria de casos) de sua autoria, o CaseMaker, que recomenda a adição de um caso com base em sua contribuição para cobertura do sistema. Ou seja, ele recomenda aqueles casos que o sistema RBC atualmente é incapaz de resolver e o autor humano decide quais casos recomendados serão adicionados na biblioteca. Os casos apresentados são listados na ordem de sua contribuição para cobertura do espaço de solução.

MULLINS e SMYTH (2001) mostram técnicas de visualização de bases de conhecimento que auxiliam usuários a entender a natural complexidade dos casos apresentados e a adicionar novos casos às bases. O Sistema de Recuperação de Casos MacRad é uma dessas ferramentas para assistência no processo de diagnóstico radiológico: os casos contêm imagens médicas escaneadas como parte de suas estruturas de solução. O DRAMA é um sistema de auxílio ao desenho para indústria aeroespacial. O CADRE é uma ferramenta de desenho arquitetônico usada na criação de novos desenhos baseados em antigos.

- **RBC em comércio eletrônico:** temos várias pesquisas em RBC que mostram soluções novas e criativas para problemas encontrados no comércio eletrônico, como a dificuldade dos compradores em encontrar o produto procurado em meio à gigantesca quantidade de ofertas da WWW.

BRIDGE (2001) descreve em seu artigo um sistema de recomendação de produtos baseado em conteúdo. Neste sistema cada produto é descrito como um caso e estes são selecionados de acordo com a aproximação com as descrições das preferências dos consumidores. O catálogo de produtos é visto como a base de casos.

ROTH-BERGHOFER (2001) mostra uma aplicação com raciocínio baseado em casos para comércio eletrônico usando o framework orange. Esta aplicação auxilia o usuário na tarefa de encontrar anúncios on-line de veículos. Uma página com informações de busca é preenchida e o sistema busca os casos (anúncios) que mais se assemelham com aquele desejado.

CUNNINGHAM, et al. (2001) trazem em seu artigo a descrição do projeto WEBSSELL, que visa desenvolver a próxima geração do suporte inteligente para vendas on-line. Este sistema usa técnicas de RBC, árvore de decisão baseada em sistemas de protocolos, perfil de usuários e recomendações cooperativas para produzir assistentes virtuais de vendas que elicitam desejos de usuários e identificam produtos e serviços que atendam esses desejos.

- **Soft Computing usando Raciocínio Baseado em Casos:** o paradigma RBC pode ser visto combinado com outras áreas da IA visando uma melhor adaptação a um

dado problema, ou na tentativa de uma nova abordagem a antigos desafios, como a previsão do tempo.

HANSEN e RIORDAN (2001) apresentam um trabalho que, segundo eles, é a única linha de pesquisa que combina o raciocínio baseado em casos com a teoria fuzzy para criar uma aplicação de previsão do tempo. Este sistema foi batizado de WIND-1, que foi proposto, implementado e testado. Os testes consistiam na previsão de 6 horas para teto de nuvens e visibilidade num aeroporto, dado um banco de dados de 300.000 horas consecutivas de observação do tempo no aeroporto (36 anos de registro). A acuracidade das previsões foi comparada com a técnica padrão de previsão. Nas simulações reais, WIND-1 foi, segundo seus autores, significativamente mais acurado.

SOVAT e CARVALHO (2001) propõem o uso de redes neurais artificiais para fazer as etapas de recuperação e adaptação de casos num sistema RBC. Este mecanismo foi necessário devido à natureza dos casos tratados pelo sistema RBC desenhado pelos autores. Cada caso representa uma RNA desenhada por um perito para resolver um problema em particular. Este trabalho lida com o problema de selecionar e adaptar esses casos. Esses casos podem ser usados como apoio para o desenho de RNAs que tratem de novos problemas.

MALEK e KANAWATI (2000) apresentam uma aplicação chamada de CoWing que faz uma busca nos arquivos favoritos (bookmarks) de usuários do sistema para identificar grupos de pessoas que compartilham interesse pelo mesmo tipo de informação, podendo criar então comunidades virtuais on-line. Um agente pessoal, chamado wing observa o comportamento do usuário ao gerenciar seu arquivo de favoritos. Cada agente wing implementa um paradigma híbrido neural/RBC que aprende como o usuário classifica seus bookmarks.

4. Raciocínio Baseado em Casos Aplicado a um Sistema de Auxílio ao Diagnóstico de Problemas em Computadores

4.1. Motivação

No início do desenvolvimento do presente trabalho, a posse de um grande banco de dados de problemas resolvidos em computadores norteou a busca de um foco fechado dentro do vasto campo da IA para a dissertação. Devido às características desses casos resolvidos, que serão estudadas adiante, a escolha do RBC aplicado em um sistema especialista se mostrou muito lógica e coerente.

O atual estágio de desenvolvimento da Inteligência Artificial e a facilidade de acesso a um grande poder de processamento, impensável até poucos anos atrás, nos leva a repensar a questão da disseminação do conhecimento. Os Sistemas Especialistas têm o poder de transferir conhecimentos de especialistas em qualquer área desejada para usuários leigos, tomando decisões ou apoiando seus usuários a tomá-las. Conforme TEIXEIRA (2000), o conhecimento pode ser preservado e transmitido a outros profissionais da área, e desse modo melhorar a formação de uma classe profissional.

Grande parte do conhecimento necessário para resolver um problema atual se encontra em experiências anteriores assemelhadas. Não é necessário se reinventar a roda a cada novo caso que se apresenta. A formação de um técnico que resolva satisfatoriamente toda a gama de problemas surgidos em microcomputadores demanda um grande investimento em tempo e dinheiro, e um sistema especialista agilizaria o processo.

Sem contar o fato que quando um técnico experiente, um especialista, sai da empresa onde trabalha ele leva junto consigo seus conhecimentos fazendo com que novos funcionários comecem do zero novamente, o que não ocorre se esses conhecimentos estiverem sistematizados num SE. YAZDANI (1986) afirma que os especialistas humanos são raros e nem sempre são dignos de confiança, requerem pagamento continuamente ao longo do tempo para usar seus conhecimentos, e morrem – levando seu saber.

Outra motivação para o desenvolvimento de um SE é a rápida disseminação de novas soluções. Quando nenhum dos casos presentes na base de conhecimento do SE servir e o usuário descobrir a solução por conta própria, este poderá lançar seu novo caso – se tiver permissão para tal – e todos demais usuários a usarão a partir daí.

4.2. Aquisição do Conhecimento: o Banco de Dados Disponível

A aquisição de conhecimento, num sistema que usa o paradigma de raciocínio baseado em casos, consiste nas formas que o sistema vai usar para alimentar sua base de casos, aumentando assim as chances de que um caso novo apresentado seja resolvido a partir daqueles armazenados. As formas de aquisição de conhecimento podem ser:

automática sem supervisão, automática com supervisão e manual. A aquisição automática é assim chamada quando o próprio sistema conta com um módulo que recolhe informações do ambiente, através de sensores, e quando detecta que o conjunto de dados recolhidos difere dos casos já armazenados este novo caso é armazenado diretamente na base de casos. Se essa armazenagem é feita sem consultar usuários humanos ela é automática, caso contrário ela é supervisionada. A armazenagem manual é aquela onde a responsabilidade por coletar, organizar, transformar e/ou guardar os dados na base é dos usuários humanos. No nosso sistema proposto a forma de aquisição de conhecimento é manual, com dois momentos distintos: a filtragem, organização e transformação dos dados que já estão armazenados em outro formato num primeiro momento, e o lançamento de novos casos a partir da interação com os usuários do sistema num segundo momento.

O primeiro momento será elucidado a seguir, com a apresentação do banco de dados disponível e com a metodologia a ser adotada para filtrar, organizar e transformar esses dados de modo a formar a base de casos do SE.

4.3. Apresentação do Banco de Dados

Para a criação da base de casos dispõe-se de um banco de dados inicial de 10804 ordens de serviço de uma empresa de assistência técnica de informática, da qual o autor do presente trabalho é sócio-proprietário, descrevendo o problema que o equipamento apresenta e a solução adotada para resolver o problema. Essas ordens de serviço referem-se ao período de setembro de 1998, quando foi implantado um sistema de armazenamento das OS, a julho de 2001, quando começou o trabalho de organização e filtragem com vistas ao desenvolvimento do protótipo de SE.

Logo a seguir vemos o leiaute da OS, e nas seções seguintes a descrição de cada um dos campos da mesma. Como podemos notar a maioria das informações que interessam ao desenvolvimento do SE são escritas em linguagem natural, sujeitas a todos os tipos de não-padronização imagináveis. Como a grande maioria das ordens de serviço é preenchida por uma funcionária de formação inadequada, tanto do ponto de vista dos conhecimentos em língua portuguesa quanto da formação técnica e domínio do glossário dos termos de informática, que recebe os equipamentos, é grande a incidência de erros sintáticos e semânticos no preenchimento dos campos. Também é freqüente o uso de termos sinônimos na descrição de um mesmo defeito ou de um equipamento, dificultando a posterior separação dos casos. Um equipamento como um disco winchester pode aparecer também descrito como HD, WT, disco rígido ou winchester. Este é mais um desafio a ser superado na análise do banco de dados e na formação da base de casos.

O exemplo mostrado é de uma ordem de serviço real, extraída do banco de dados original, onde somente foram alteradas algumas informações com a finalidade de manter sigilo de dados pessoais, como os dados de identificação do cliente, número da OS, sua data, hora e valor do serviço.

Compunew Computadores e Assistência Técnica Ltda. Upgrades - Redes Locais - Assistência a Micros Av. América, 196. Fone/Fax: (055) 3512-5214 CGC: 68.846.500/0001-82 IE: 110/0053740		Atendimento (X) Balcao () Externo () Misto	N.OS: 000000 01/01/99 10:10:10 USUÁRIO	
Cliente : Fulano de Tal da Silva (9999) Endereço: Rua não sei qual S/N Complem.: Bairro das Dores Partidas - Cidade sem Rumo/RS Telefone: (99) 999-9999 Fax/Cel: (99) 999-9999 Contato: Fulano de Tal				
Serviço a ser prestado / Problema que o equipamento apresenta Puxa folha, mas não imprime e fica piscando luz de cima				
Equipamentos e acessórios recebidos			Volt	
Impressora HP 692C NS: BR60XG109A com fonte e cartuchos			110	
Descrição dos serviços prestados	Técnico	Horas	TA	Preço
Detecção de defeito no circuito do cartucho colorido	NomeTec	1.00	4	00,00
Peças e componentes substituídos			Preço	
01 cartucho color original HP692C			00,00	
		Total	:	00,00
Visto do Entregador		Dt.Fecha: 99/99/99		
Visto do Cliente				
Compunew Computadores e Assistencia Técnica Ltda. Upgrades - Redes Locais - Assistência a Micros Av. América, 196. Fone/Fax: (055) 512-5214 Cliente : Fulano de Tal da Silva (9999) Puxa folha, mas não imprime e fica piscando luz de cima		N.OS: 123456 01/01/99 10:10:10		

Figura 4.1. Leiaute da Ordem de Serviço

4.3.1. O Cabeçalho e o Rodapé da Ordem de Serviço

O cabeçalho da OS traz dados de identificação da empresa prestadora de serviços e do usuário do equipamento com problema, nome de quem preencheu a OS, além da data e hora do preenchimento. O rodapé traz informações semelhantes, tendo ainda um espaço para assinatura das partes envolvidas.

Para o protótipo de sistema especialista proposto estas partes da OS não trazem nenhum tipo de informação relevante. Num posterior desenvolvimento poderia se fazer estatísticas a respeito do tempo médio de solução de um determinado problema a partir da data e hora de abertura e fechamento de uma OS, ou ainda a respeito da reincidência de um determinado problema com um mesmo usuário. Essas questões voltam a ser discutidas no capítulo 6.

4.3.2 A Seção *Serviço a ser prestado / Problema que o equipamento apresenta*

A primeira parte preenchida no corpo de nosso modelo de ordem de serviço é a descrição do problema que o equipamento apresenta, em linguagem natural. São muito comuns descrições do tipo: não liga, impressora não puxa o papel, cpu¹ liga mas não dá sinal de vídeo, remover vírus, instalar processador de texto, reinstalar windows, etc.

Um dos “ruídos” mais frequentes nos dados desta seção é a presença de nomes, endereços e telefones de clientes. Quando um equipamento é deixado por uma pessoa que não possui cadastro e esta tem muita pressa ou não possui todos os dados para preenchimento do cadastro (CPF, telefone, etc.), a OS é emitida com o nome *Cliente sem Cadastro* e na primeira linha desta seção colocam-se alguns dados da pessoa para futura identificação.

O próprio título dessa seção da OS mostra que esses documentos não são emitidos somente quando existe um problema, mas também quando o equipamento está em perfeitas condições e seu usuário deseja somente uma manutenção preventiva, a instalação de algum software novo ou algum outro tipo de configuração que não denota defeito.

A principal filtragem a ser executada no banco de dados antes de transferir suas informações para a base de casos é justamente a eliminação daquelas OS que não representam equipamentos com defeitos, e a descrição dos critérios dessa seleção é feita numa seção mais adiante neste mesmo capítulo.

4.3.3 A Seção *Equipamentos e acessórios recebidos*

Essa parte da ordem de serviço é, daquelas preenchidas em linguagem natural, a que pode se dizer ser mais estruturada, ou mais fácil de ser processada, pois não contém ruídos como dados dos clientes, e quase sempre segue a ordem: equipamento, número

¹ CPU deve ser entendida neste texto como o gabinete do microcomputador com todo o hardware interno.

de série e acessórios. Uma das dificuldades de processamento é o fato de que muitos equipamentos não são descritos por seu nome genérico, mas sim por sua marca. Desta forma, muitas vezes encontramos OS com o equipamento descrito como Epson Stylus Color, por exemplo, sem citar a palavra impressora.

A escrita do número de série dos equipamentos também é um fator complicador, já que muitos são compostos de letras e números, ou somente letras, podendo confundir o filtro separador de palavras que os detectaria como um nome de equipamento e não um número de série, mas isso já é assunto para o capítulo sobre ordenamento de casos.

4.3.4 A Seção *Descrição dos serviços prestados*

Como o próprio nome já diz, nessa seção o técnico que executou o reparo no equipamento descreve o que foi feito visando chegar na solução. Além disso, também são preenchidos o nome do técnico, o tempo em horas que o conserto levou, o tipo de atendimento (1=Conserto em garantia de fábrica, 2=Conserto em garantia do revendedor, 3=Orçamento, 4=Conserto de hardware fora de garantia, 5=Conserto de software fora de garantia) e o preço do conserto.

Entrevistas com os técnicos mostraram que nem sempre tudo que foi feito é descrito na OS, basicamente porque num pequeno percentual dos casos ocorre um dos três motivos abaixo:

- a descrição pode se tornar demasiadamente extensa tomando muito tempo do técnico que a preenche e da secretária que a digita posteriormente;
- a descrição muitas vezes conteria termos de jargão técnico ininteligíveis para os donos dos equipamentos;
- alguns procedimentos de solução de problemas são considerados “segredos técnicos” que se tornariam de domínio público e não gerariam mais serviços daquela natureza.

Obviamente este procedimento é prejudicial para a formação da base de casos, já que a relação direta [descrição do defeito⇒descrição da solução] pode não ser verdadeira em todos os casos. Para contornar essa situação a base de casos deve sofrer uma revisão de especialistas após sua formação visando à eliminação de incoerências, que devem ocorrer, repetindo, em um pequeno percentual.

Outras ordens de serviço simplesmente não contêm descrição nenhuma acerca dos serviços prestados, por não ter sido feito nenhum serviço mesmo. Essas ordens serão descartadas do banco de dados durante o processo de filtragem, como veremos mais adiante. Essa situação de falta de descrição de serviços pode acontecer em quatro situações distintas:

- quando a solução não foi encontrada pelos técnicos;
- quando a solução existe, é conhecida, mas não foi possível aplicá-la por razões como falta de peças ou componentes para substituição;
- quando a solução é conhecida e viável de ser aplicada mas o cliente não aprovou o orçamento;

- quando o técnico trocou a peça ou componente e somente preencheu a seção peças e componentes substituídos, não descrevendo o serviço feito.

4.3.5 A Seção *Peças e componentes substituídos*

Quando a solução de um defeito em um equipamento exige a substituição de alguma peça ou componente, a descrição do mesmo aparece nesta seção. Um problema sério para a filtragem ocorre quando eventualmente um técnico resolve o problema trocando uma peça e não escreve nada na descrição dos serviços prestados – às vezes isso acontece. Na base de casos a sugestão de solução para esse defeito deverá ser a substituição da peça/componente apesar de não termos descrição de serviços prestados.

4.4. A Filtragem dos Problemas Reais e com Solução

Conforme foi visto anteriormente, no item 4.3.2., o banco de dados que servirá de fonte para a base de casos não contém somente ordens de serviço descrevendo a solução de problemas em equipamentos de informática, mas também contém a descrição de serviços de manutenção preventiva, a instalação de softwares, ou a configuração de algum componente que não está com defeito.

Dito isso, a seguinte questão se impõe: como fazer para separar as ordens de serviço que descrevem problemas, que são as que nos interessam no desenvolvimento deste trabalho, daquelas que não tratam problemas? Qual o melhor método de filtragem, visto que ler e classificar manualmente uma a uma mais de dez mil ordens disponíveis não seria viável?

Começamos pensando em como a tarefa seria feita caso fosse adotada a solução de separação manual. A primeira seção da OS seria lida, aquela que traz por título *Serviço a ser prestado / Problema que o equipamento apresenta*. A partir do texto desta seção, sem precisar do restante dos dados da OS, parece lógico que já seria possível fazer a classificação. De um lado apareceriam as ordens com descrições do tipo: cpu não liga, impressora não puxa o papel, oscila imagem do monitor. De outro lado apareceriam ordens com descrições como: instalar processador de texto, reinstalar windows, configurar scanner. A partir de conversas com 5 técnicos especialistas, foi criada uma lista das palavras que mais provavelmente apareceriam na descrição de um equipamento com defeito. Várias das palavras relacionadas continham o mesmo radical, o que levou a uma condensação, reduzindo palavras como tranca, trancou, trancava ao radical tranc. Também outras palavras que são escritas de forma errada com frequência, como oscilar, foram procuradas nas duas formas comumente escritas: oscila e ocila.

O resultado destas entrevistas e a compilação final da lista de palavras descritivas de problemas aparecem logo a seguir na tabela 4.1.

Palavra ou radical
Barulho
Borr
Chuvisc
Consert
Corrig
Erro
Esquent
Falh
Funciona
Imagem
Nao ²
Ocila
Oscila
Pendura
Problema
Quebr
Queim
Recuper
Reset
Tela azul
Tranc
Trav
Trem
Troc
Vaz
220

Tabela 4.1. Palavras que indicam equipamentos com problemas

Um programa criado para esse fim fez uma varredura na primeira seção do banco de dados de ordens de serviço buscando as palavras constantes na tabela anteriormente apresentada e separou as mesmas em 2 arquivos distintos: o de ordens descritivas de problemas, aquelas onde pelo menos uma das palavras da lista apareceu; e o de ordens descritivas de outros serviços, onde nenhuma das palavras da lista constou. Após a realização desta varredura, o banco de dados inicial de mais de dez mil ordens ficou dividido conforme a tabela 4.2.

Nº de ordens descrevendo problemas/defeitos	5.084
Nº de ordens descrevendo outros serviços	5.720
TOTAL DE ORDENS DE SERVIÇO	10.804

Tabela 4.2. Divisão do Banco de Dados de Ordens após a Filtragem.

² Deve-se observar que se buscou no texto a palavra *nao* sem o acento til, pois o banco de dados original tem todas palavras armazenadas sem acentuação. Esta situação se repete ao longo do texto.

Visando conferir o grau de precisão do método de filtragem adotado, as primeiras 100 ordens do banco de dados foram conferidas uma a uma manualmente, comparando-se o resultado obtido desta maneira com aquele obtido automaticamente pelo programa. O resultado desta comparação aparece na tabela 4.3.

	Método de Filtragem	
	Manual	Automático
Nº de ordens com problema/defeito	72	73
Nº de ordens com outros serviços	28	27
TOTAL	100	100
Nº de ordens em que os dois métodos coincidiram no resultado da filtragem: 91		

Tabela 4.3. Comparação dos métodos de filtragem manual e automático.

Ressaltamos ainda que das 9 ordens classificadas de forma errada pelo método automático, 5 delas foram classificadas como ordens com problema/defeito quando eram ordens de outros serviços, e 4 delas foram classificadas como OS de outros serviços quando eram ordens com problemas/defeitos.

Podemos então estimar por esta amostragem apresentada, que nosso método automático de filtragem de OS tem cerca de 91% de precisão, o que permite um cálculo de que, das 5.084 ordens selecionadas para compor a base de casos, mais de 4.600 devem conter efetivamente a descrição de equipamentos com problema.

Além de filtrar as ordens do banco de dados que se referem efetivamente a problemas, também devemos nos preocupar em manter no BD somente as que têm solução efetiva, eliminando aquelas que não trazem descrição de serviço efetuado, ou que não trazem substituição de peças ou componentes, que poderia ser entendido como uma solução.

Esse passo seguinte da filtragem foi realizado, separando e eliminando do banco de O.S. aquelas que traziam em branco as seções *Descrição dos serviços prestados* e *Peças e componentes substituídos*. O resultado desta segunda filtragem, realizada somente sobre as 5.084 ordens restantes do primeiro passo aparece na tabela 4.4.

Nº de ordens após a primeira filtragem	5.084
Nº de ordens com as duas últimas seções em branco	1.118
Nº de ordens após a segunda filtragem	3.966

Tabela 4.4. Composição do banco de ordens após a segunda filtragem.

4.5. O Ordenamento dos Casos

Agora que temos em mãos as ordens de serviço que irão compor a base de casos, deve ser desenvolvida uma forma de ordenamento das mesmas visando agrupar aquelas que se referem a problemas semelhantes, facilitando a posterior localização de soluções na base de casos. A princípio podemos pensar em três diferentes abordagens para o problema de ordenamento de casos semelhantes: por descrição do problema, por descrição da solução adotada, e por equipamento que apresenta o defeito.

Caso fosse adotada a classificação por descrição do problema, aparentemente a mais lógica, o leque de possibilidades de solução tenderia, em alguns casos, a se abrir demasiadamente. Por exemplo, uma descrição de problema freqüente é “não liga”. Essa descrição pode estar se referindo a um monitor de vídeo, uma CPU, uma impressora, um estabilizador, etc., dificultando o encontro de uma solução eficiente para tantas situações possíveis.

Já se fosse adotada a classificação por descrição da solução adotada, um paradoxo seria criado, pois o usuário teria que saber a solução para o problema de antemão para encontrar o caso pretendido, não fazendo sentido então o uso de um sistema especialista que se propõe justamente a encontrar para ele o caso mais semelhante da base de casos e então dar a solução.

Parece então ter sobrado a classificação por equipamento que apresenta o defeito. Mas esse tipo de ordenamento também traz embutido em si o problema de abrir inúmeras possibilidades, já que uma CPU, por exemplo, pode ter centenas de defeitos, assim como qualquer outro tipo de equipamento. Propomos então uma solução mista: que classifique inicialmente os casos pelo equipamento defeituoso, e dentro de cada equipamento pela descrição do problema apresentado. Existe uma exceção que deve ser prevista nesta ordenação, que são os chamados defeitos de software, que abrange as configurações incorretas, a falta ou corrupção de arquivos, os conflitos de versões e outros problemas típicos. Nesse caso não temos nenhum equipamento defeituoso propriamente dito, mas a seção *Equipamentos e acessórios recebidos* terá com certeza a presença de uma CPU. O tratamento para esta questão será esclarecido no item 4.5.1.

Como nosso banco de dados de OS não tem um ordenamento prévio pelos critérios que acabamos de definir, foi criado um programa que classifica palavra por palavra os textos da seção *Equipamentos e acessórios recebidos* das OS. Ordenando o número de vezes que cada palavra aparece teremos subsídios para começar o cruzamento com a seção *Problema que o equipamento apresenta* e finalmente começar a criação da base de casos.

4.5.1. Os Equipamentos Mais Frequentes

O programa que classificou por número de ocorrências as palavras da seção *Equipamentos e acessórios recebidos* das OS não levou em conta sinônimos já mencionados anteriormente no trabalho. Desta forma, as palavras *micro*, *CPU*, *computador* e *microcomputador* foram contabilizadas como sendo equipamentos distintos. Também não foram levadas em conta as palavras denotando marcas

comerciais como sinônimos de equipamentos. Assim, os técnicos, especialistas e usuários sabem que o equipamento descrito como *Epson Stylus Color* se refere a uma impressora, mas o programa contou como 3 palavras distintas e nenhuma delas como sendo impressora.

O resultado parcial dessa primeira triagem aparece na tabela 4.4., onde são listadas as 50 palavras que mais apareceram, de um total de 4.517 palavras diferentes, totalizando 30.598 ocorrências nesta seção. Lembramos que esses números se referem somente às 3.966 ordens selecionadas para compor a base de casos, e não às mais de 10.000 ordens iniciais.

A palavra *NS* é a que mais ocorre pois todo equipamento tem seu número de série anotado na OS, e a letra *C* isolada aparece em segundo lugar pois sempre que o equipamento está acompanhado de acessórios escreve-se *c/cabo*, *c/cartuchos*, *c/capa*, etc.

Seqüência	Palavra	Nº.Ocorrências	Seqüência	Palavra	Nº.Ocorrências
1	Ns	3708	26	Fonte	173
2	C	1223	27	Alfa	171
3	Cpu	1136	28	Compaq	171
4	Imp	953	29	Manopla	168
5	Micro	775	30	Lx	164
6	Hp	759	31	Megatropic	156
7	Impressora	650	32	Uis	155
8	E	614	33	Break	143
9	S1	594	34	Digital	141
10	Epson	509	35	Paralelo	141
11	Monitor	486	36	Teclado	133
12	Cartuchos	483	37	N5455	131
13	Fita	394	38	Transformador	130
14	Cabo	357	39	Mouse	126
15	De	311	40	600	114
16	Tampa	289	41	Marca	113
17	Com	286	42	Stylus	112
18	Sem	255	43	Forca	109
19	No1w6x01069	247	44	Cd	102
20	Compunew	236	45	680c	95
21	Nobreak	208	46	Drive	89
22	Cart	191	47	Color14	85
23	300	191	48	Emilia	83
24	692c	187	49	M	79
25	Olivetti	181	50	Pc	79

Tabela 4.5. Palavras com maior ocorrência na seção Equipamentos e acessórios recebidos

Pela simples leitura da tabela apresentada fica claro que deve ser feito um trabalho de limpeza e agrupamento, permanecendo apenas as palavras que se referem efetivamente a equipamentos e se eliminando marcas comerciais, números de série,

erros ortográficos e outras palavras que não contribuem para a formação da base de conhecimentos.

Esse trabalho de limpeza e agrupamento da relação dos equipamentos foi feito a pedido do autor por 5 técnicos especialistas em informática resultando então na tabela 4.6., que pode ser considerada a relação de todos diferentes tipos de equipamentos que foram consertados e descritos nas quase 4.000 ordens de serviço usadas no desenvolvimento deste trabalho. O número de vezes que cada equipamento apareceu nas OS não foi levantado com precisão pois os sinônimos são vários para cada equipamento e o fato de alguns serem descritos por mais de uma palavra traria a necessidade de uma contagem manual. Porém, a ordem com que as palavras aparecem na tabela é de acordo com o número de ocorrências do principal sinônimo de cada equipamento, o que já dá uma boa idéia a respeito dos equipamentos mais frequentes, afirmação essa confirmada pelos técnicos especialistas quando confrontados com a tabela.

Seqüência	Equipamentos
1	Cpu
2	Impressora
3	Monitor
4	Nobreak
5	Fonte
6	Teclado
7	Mouse
8	Drive de Disquete
9	Fax
10	Estabilizador
11	Winchester
12	Notebook
13	Scanner
14	Cabeça de Impressão
15	Zip Drive
16	Palmtop
17	Hub
18	Cd-Rom
19	Modem
20	Caixas de Som
21	Memoria
22	Bloco Movel
23	Joystick
24	Fita Dat
25	Microfone
26	Cooler
27	Motherboard

Tabela 4.6. Equipamentos e acessórios das ordens de serviço após filtragem.

Observando a tabela acima podemos perceber que muitos dos equipamentos listados são componentes menores de outros equipamentos da tabela, alguns da CPU (memória, cooler, motherboard, drives, etc.), outros das impressoras (bloco móvel, cabeça de impressão) e assim por diante. Para facilitar ao usuário do sistema especialista a determinação do equipamento ou acessório com problema foi pensada numa estrutura em árvore hierárquica onde aparecesse de forma clara que componentes fazem parte de cada equipamento e onde o usuário pudesse apontar de maneira unívoca o que está apresentando defeito. Nesta árvore seriam também encaixados os defeitos de software, citados como exceção no item anterior do texto, aparecendo como uma ramificação dos defeitos de CPU naquela que foi batizada como árvore hardware/software e doravante citada como árvore HS.

Visando uma melhor delimitação dos problemas cobertos pelo SE, foram eliminadas algumas das entradas da tabela 4.6 em sua transição para a criação da árvore HS. Alguns itens foram eliminados por seu caráter descartável, sendo que na maior parte dos casos vale mais a pena trocar o equipamento por um novo do que consertá-lo, tais como: teclado, mouse, joystick, caixas de som e microfone. Os itens notebook e palmtop foram eliminados porque seus defeitos podem ser encaixados em um dos outros itens da árvore (CPU ou monitor). O item fax foi eliminado por não ser considerado propriamente um equipamento de informática. Uma nova triagem de ordens será feita mais adiante para refletir esta delimitação do universo dos problemas cobertos pelo SE.

Esta nova relação de equipamentos com uma sugestão de divisão hierárquica aparece na tabela 4.7 e não se pretende que seja definitiva, mas pelo contrário, seja apenas o ponto de partida dentro do SE para a inclusão de novos equipamentos e para um futuro refinamento.

EQUIPAMENTOS	
CPU	
	Ligamento
	Armazenamento
	Windows
	Internet
	Outros
Impressora	
	Matricial
	Jato de Tinta
	Laser
Monitor	
	Ligamento
	Botões/Painel
	Imagem
No Break	
Estabilizador	
Scanner	

Tabela 4.7. A primeira versão da árvore HS.

Quando o usuário do SE interagir com o sistema, com frequência ele irá se deparar com a questão de não saber delimitar exatamente onde está o defeito, optando então por um nível mais acima na árvore HS e esperando então que dentre as alternativas de problemas apresentados ele encontre o que melhor se encaixe no seu caso. O SE deve permitir essa atitude, mas com a ressalva que a automatização do processo de casamento do melhor caso da base de casos com o novo caso apresentado fica comprometida.

4.5.2. Os Problemas Mais Frequentes

Assim como foi feito com os equipamentos, criou-se um programa para classificar as palavras com maior ocorrência na seção de problemas/defeitos. O resultado parcial dessa primeira triagem aparece na tabela 4.7., onde são listadas as 50 palavras que mais apareceram, de um total de 3.681 palavras diferentes, totalizando 33.437 ocorrências nesta seção. Lembramos que esses números se referem somente às 3.966 ordens selecionadas para compor a base de casos, e não às mais de 10.000 ordens iniciais.

Seqüência	Palavra	Nº Ocorrências
1	Não	3241
2	E	750
3	Declara	671
4	Liga	617
5	Imprime	478
6	Micro	473
7	Acessa	462
8	Problema	408
9	Windows	398
10	Da	387
11	Do	370
12	Impressao	354
13	No	347
14	Funciona	332
15	Verificar	282
16	Trancada	257
17	Barulho	249
18	Trocar	244
19	Para	241
20	Nacional	239
21	Faz	227
22	Papel	220
23	Error	218
24	Falha	213
25	Obedece	193

Seqüência	Palavra	Nº Ocorrências
26	Impressora	192
27	Coml	184
28	As	182
29	Placa	179
30	Esta	175
31	Drive	174
32	Imagem	172
33	Monitor	170
34	Orcamento	169
35	Veze	166
36	Def	166
37	Rede	163
38	Cd	160
39	512	159
40	Oab	156
41	Fazer	153
42	Hd	148
43	A20	148
44	Limpeza	144
45	Inicializa	142
46	Ao	142
47	Passar	137
48	Tem	136
49	Emilia	136
50	Trava	133

Tabela 4.8. Palavras com maior ocorrência na seção *Problema que o equipamento apresenta*

A palavra *não* é a que mais ocorre, com grande diferença sobre as demais, o que motivou a criação da tabela 4.8, onde aparecem os desdobramentos dos casos onde a palavra *não* aparece, juntamente com a palavra que a acompanha. Somente foram listados os primeiros 50 *nãos* compostos, de um total de 184 diferentes combinações encontradas no banco de dados.

Seqüência	Palavras	Nº Ocorrências
1	Nao Liga	565
2	Nao Acessa	462
3	Nao Imprime	409
4	Nao Funciona	311
5	Nao Inicializa	136
6	Nao Segura	103
7	Nao Tem	96
8	Nao Traciona	94
9	Nao Carrega	87
10	Nao Puxa	86
11	Nao Le	81
12	Nao Comunica	69
13	Nao Abre	53
14	Nao Esta	50
15	Nao Da	43
16	Nao Reconhece	34
17	Nao Entra	32
18	Nao Detecta	28
19	Nao Conecta	26
20	Nao Passaram	25
21	Nao Aparece	24
22	Nao Faz	21
23	Nao Funcionam	18
24	Nao Aceita	15
25	Nao Roda	15

Seqüência	Palavras	Nº Ocorrências
26	Nao Sai	15
27	Nao Desliga	13
28	Nao Eh	12
29	Nao Grava	10
30	Nao Responde	10
31	Nao Consegue	9
32	Nao Ligam	9
33	Nao Anda	8
34	Nao Aciona	6
35	Nao Disca	6
36	Nao Instala	6
37	Nao Recebe	6
38	Nao Clica	5
39	Nao Comunicam	5
40	Nao Estao	5
41	Nao Imprimem	5
42	Nao Imprimia	5
43	Nao Navega	5
44	Nao Obedece	5
45	Nao Carega	4
46	Nao Completa	4
47	Nao Escaneia	4
48	Nao Formata	4
49	Nao Transmite	4
50	Nao Apagar	3

Tabela 4.9. Amostra das combinações da palavra *não*.

4.6. Montando a Base de Casos

Dado o fato que no item 4.5 propusemos uma classificação dos casos inicialmente pelo equipamento defeituoso, e dentro de cada equipamento pela descrição do problema apresentado, e como a análise das tabelas com palavras descritivas de problemas (tabelas 4.8 e 4.9) não é elucidativa, fez-se então o cruzamento dos equipamentos cobertos pelo SE com os defeitos encontrados. Inicialmente dividiram-se as ordens de serviço de acordo com o tipo de equipamento que elas traziam, e o resultado aparece na tabela 4.10.

Para realização desta separação, foram buscadas ordens que tivessem na descrição do equipamento uma das palavras sinônimas constantes da tabela. O total de ordens encontradas é superior as 3.966 OS do banco de dados de trabalho pois várias ordens contam com mais de um equipamento.

Equipamento	Sinônimos buscados	Nº de Ordens
CPU	Micro, computador, PC	1.540
Monitor	Video, Tela	386
Impressora	Epson, HP, Olivetti, Canon, Imp	2.347
No-Break	Nobreak	159
Estabilizador		43
Scanner		21
TOTAL		4.496

Tabela 4.10. Divisão das OS por equipamento

As ordens relativas a scanners, estabilizadores e no-breaks, por serem em menor quantidade, foram classificadas manualmente de acordo com os diferentes tipos de problemas apresentados. As demais ordens passaram por mais uma fase de separação por palavras mais freqüentes em três tabelas diferentes, uma para CPUs, uma para impressoras e outra para monitores e então foram extraídos os defeitos mais comuns para a montagem da base de casos. O anexo 1 traz a relação dos defeitos encontrados em cada um dos 6 diferentes tipos de equipamentos cobertos pelo sistema e as palavras-chave relacionadas a cada defeito.

4.7. Apresentação da Proposta de Algoritmo de Casamento

No desenvolvimento de Sistemas Especialistas que usam o RBC, como já vimos no capítulo 3, *casamento* é o termo que designa a proximidade de um caso armazenado na base com aquele novo caso apresentado pelo usuário. Quanto mais afinidade entre um caso da base e o novo caso, se diz ser melhor o casamento.

O algoritmo de casamento é portanto, a forma de calcular a proximidade entre dois casos, estabelecendo diferentes graus de casamento, ou distâncias, para diferentes pares de casos, sempre visando encontrar o par mais próximo. Esses pares sempre são formados de um lado pelo caso novo trazido pelo usuário do SE e de outro lado por um dos casos já resolvidos e armazenados na base.

Como nossas ordens de serviço que formarão a base de casos têm descrições textuais dos problemas e soluções, o algoritmo de casamento buscará palavras em comum, presentes na descrição do problema novo trazido pelo usuário e nas descrições dos problemas armazenados na base do SE. Visando uma otimização do tempo de busca, não precisando varrer toda a base de casos, e também uma maior probabilidade de acerto, o usuário indicará no início da sua interação qual é o equipamento com

problema dentre os seis tipos cobertos pelo SE (CPU, monitor, impressora, estabilizador, no-break e scanner) e, se possível ou cabível, que parte do equipamento é a defeituosa. A busca de palavras em comum ocorrerá somente nos casos pertencentes à categoria de equipamento discriminada pelo usuário e os casos semelhantes que forem encontrados serão apresentados ordenados de acordo com o grau do casamento calculado pelo algoritmo do sistema. Teoricamente a melhor solução para o problema apresentado será a primeira da lista, caso um problema semelhante já tenha sido resolvido e armazenado no SE.

O sistema conta com um dicionário das palavras descritivas de defeitos, chamadas no texto daqui para frente de PDD, armazenadas juntamente com o equipamento a que ela se refere e o grau de importância daquela palavra na descrição do problema em questão. Por exemplo, um defeito bastante comum em monitores é a oscilação da imagem. Neste caso, a palavra *oscila* é armazenada com grau de importância “2”, já que descreve o defeito de maneira clara e unívoca. Já a palavra *imagem* é armazenada com o grau de importância “1”, pois apesar de ser importante, outros defeitos de imagem existem.

Neste dicionário uma mesma palavra pode aparecer diversas vezes, se referindo a diferentes casos, e até mesmo a equipamentos diversos, e podendo assumir pesos ou graus de importância diferentes.

A figura 4.2 mostra um fluxograma descritivo de como funcionaria o algoritmo de casamento proposto e desenvolvido para o SE apresentado neste trabalho. Nesta figura assumimos que o equipamento com defeito faz parte do domínio do conhecimento do sistema e que foi encontrado pelo menos um caso semelhante. Internamente o sistema terá de tratar também essas exceções.

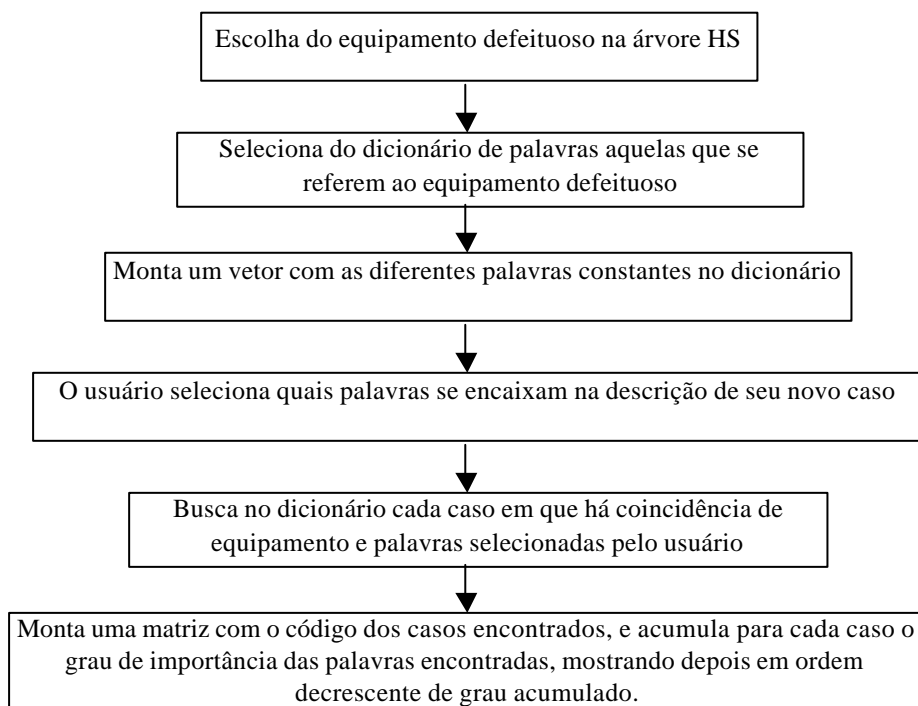


Figura 4.2. Fluxograma do algoritmo de casamento proposto.

4.7. Considerações sobre os Problemas do Domínio

Sabemos que Sistemas Especialistas, como o próprio nome diz, restringem seus conhecimentos a uma área muito particular e especializada do saber. Quanto mais restrita for esta área coberta pelo SE, também conhecida como domínio do conhecimento, mais provável será que a base de casos tenha a informação buscada pelo usuário do sistema.

Posto isso, os usuários deste protótipo apresentado devem sempre ter em mente as restrições impostas pelo domínio do conhecimento delimitado para este sistema. Somente são cobertos defeitos em 6 tipos de equipamentos: CPU's, monitores, impressoras, no-breaks, estabilizadores e scanners. Também se deve lembrar que o banco de dados que deu origem à base de casos foi gerado no período de 1998 a 2001. Assim sendo, dificilmente encontraremos soluções para equipamentos muito mais antigos ou novos que estas datas-limite, a não ser que a base de casos sofra manutenções preenchendo períodos fora deste descrito.

O perfil dos clientes da empresa de assistência técnica que forneceu o banco de dados inicial também molda os tipos de problemas cobertos pelo sistema. A maioria dos equipamentos atendidos funcionava no chamado mercado SOHO³. Dessa forma dificilmente encontraremos respostas para problemas comuns a equipamentos de mercados corporativos, de grandes empresas, como por exemplo defeitos em implementação de RAID via hardware, comuns em servidores de porte médio ou grande, mas com facilidade teremos respostas para problemas em impressoras de baixo custo.

Finalmente ressaltamos ainda a questão do engano, bastante comum, acerca de qual equipamento está efetivamente com defeito. Na maioria dos casos é fácil de diagnosticar, mas nem sempre. Por exemplo, uma impressora que não imprime leva a crer em problema de impressora, mas pode ser defeito na interface paralela do computador. Como o funcionamento de nosso protótipo se baseia em inicialmente o usuário apontar qual equipamento está com problema, um erro destes pode levar a uma frustração do algoritmo de casamento, levando o usuário a crer que o seu problema não tem caso semelhante armazenado no SE. Portanto, na dúvida deve-se marcar mais de um equipamento na primeira etapa da interação e analisar as PDD's apresentadas com bastante atenção.

³ SOHO = Small Office - Home Office. Refere-se ao mercado das pequenas empresas e escritórios e usuários domésticos de computadores.

5. Modelagem e Implementação do Sistema SADPC

Este capítulo visa apresentar o sistema especialista desenvolvido em forma de protótipo, com o intuito de mostrar na prática como funcionariam as idéias desenvolvidas ao longo do presente texto. Começamos falando das ferramentas utilizadas para o desenvolvimento do software e como o mesmo foi analisado, modelado e implementado. Posteriormente é feita a descrição de cada módulo do programa, qual sua função e como usá-lo.

5.1. Uma Visão Geral das Técnicas Adotadas

A criação do protótipo depende de algumas decisões técnicas, do tipo: escolher o método de modelagem do sistema, a plataforma de hardware e sistema operacional adotada, a linguagem de programação, etc.

A modelagem do sistema foi feita usando o método middle-out (Ricardo Pereira e Silva, 2001), começando pela identificação dos elementos do sistema, posteriormente refinando estes elementos para reduzir o grau de abstração e chegar nos atributos, métodos e algoritmos dos métodos, e chegando enfim à composição do sistema, com o desvendamento do relacionamento entre cada módulo.

A plataforma escolhida foi o hardware de computadores do tipo PC *Intel* compatível rodando o sistema operacional Windows 9X/ME, pela sua popularidade e grande aceitação.

A busca da linguagem de programação a ser usada para a criação deste sistema especialista foi norteadada pelos seguintes fatores:

- Ser popular, de grande aceitação e conhecimento do público da informática em geral, pois o código do sistema deve ser facilmente compreendido;
- Ter possibilidade de desenvolvimento em plataforma *Windows*, por ser o sistema operacional encontrado na maioria dos computadores;
- Poder ser migrado facilmente para outras plataformas caso seja necessário;
- Apresentar os recursos necessários para a representação adequada dos casos.

A partir destas orientações, a linguagem escolhida foi o *Delphi*, da *Borland*. Os arquivos de dados do sistema foram criados no formato *Paradox 7* e o desenvolvimento do sistema foi feito em um microcomputador do tipo PC, com processador AMD-K7 Duron de 800Mhz, 128Mb de memória RAM, disco winchester de 20Gb, rodando o sistema operacional Windows 98 (Segunda Edição).

5.2. Apresentação do Sistema SADPC

A tela de abertura do sistema aparece logo a seguir, na figura 5.1, onde preenchemos o apelido do usuário atual, para que o sistema saiba que se trata de um usuário final, um técnico ou um engenheiro do conhecimento.

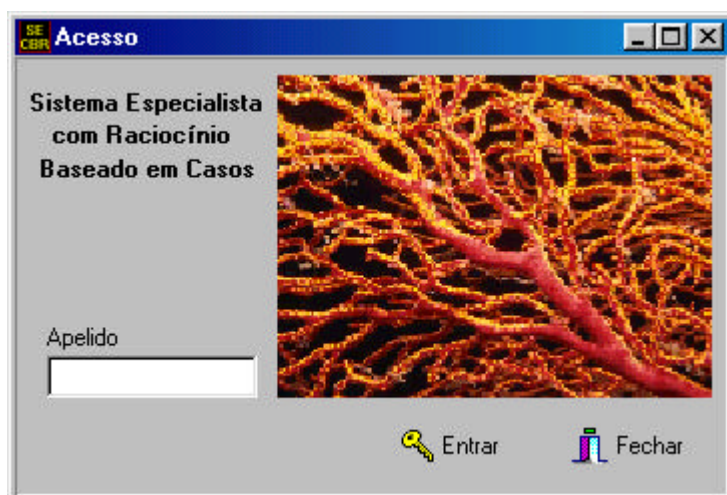


Figura 5.1. A tela de acesso ao sistema.

Em seguida, a partir do correto preenchimento de um apelido de usuário válido (cadastrado), temos a tela principal do sistema, mostrada na figura 5.2.

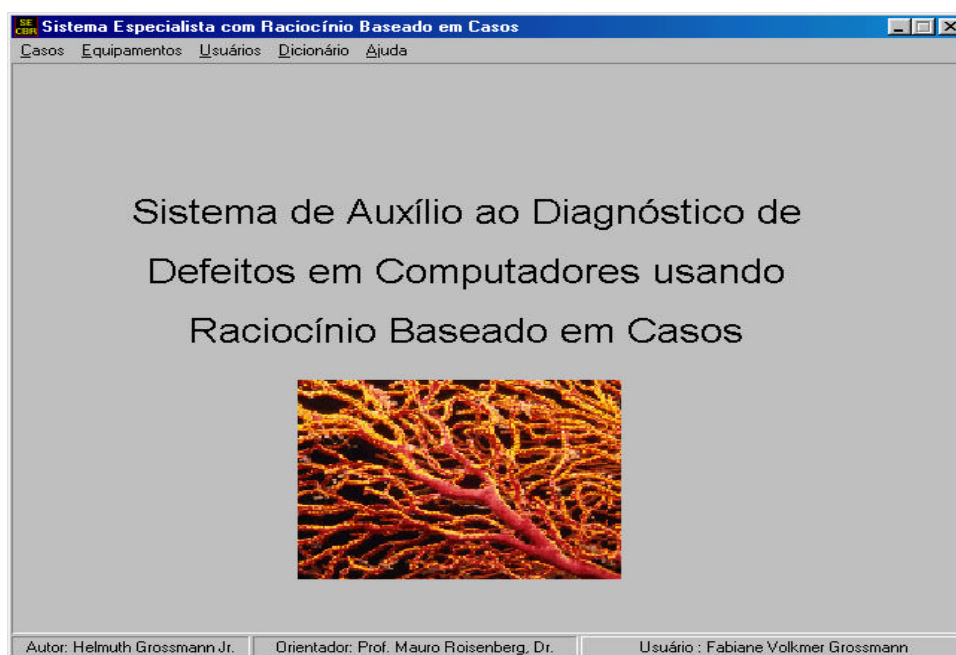


Figura 5.2. A tela principal do sistema

Nesta figura podemos distinguir de saída o menu com acesso aos 4 principais módulos do protótipo, além da ajuda. Os módulos constituintes são: a base de casos, a árvore de equipamentos, o dicionário de PDD's e o controle de usuários, correspondendo a cada módulo um arquivo de dados físico. Os três primeiros arquivos, de casos, equipamentos e o dicionário de PDD's, formam a base de conhecimento do SE, e o arquivo de usuários define como será a interação das pessoas com o sistema. A descrição do funcionamento de cada módulo e a forma como eles interagem para o funcionamento do sistema são descritos a seguir.

5.3. Os Usuários do Sistema

Por definição durante a análise e modelagem do SE, é permitido o acesso ao sistema por 3 tipos de usuários: os especialistas técnicos de informática, os engenheiros do conhecimento e os usuários leigos. Os direitos de cada tipo de usuário e o papel de cada um no uso do sistema é descrito nos tópicos pertinentes a seguir.

As informações armazenadas sobre cada usuário do sistema são: apelido, nome do usuário e tipo de usuário. A estrutura deste arquivo pode ser vista na tabela 5.1 e a chave primária de acesso deste arquivo é o campo apelido.

Nome do campo	Tipo	Tamanho
APELIDO	Alpha	6
NOME	Alpha	40
TIPO	Alpha	1

Tabela 5.1. A estrutura do arquivo de usuários.

A tela do sistema onde são feitos o cadastro e manutenção na tabela de usuários é mostrada logo a seguir, na figura 5.3.

Usuários

APELIDO: ADEMIR NOME: Ademir Squinzani TIPO: T

E = Especialista
T = Técnico
U = Usuário

Apelido	Nome Completo	Tipo
ADEMIR	Ademir Squinzani	T
ALEX	Alessandro Squinzani	U
ALICE	Alice P. da Luz	U
CHICA	Fabiane Volkmer Grossmann	E
HGJ	Helmuth Grossmann Jr	E
MARI	Mari Tânia Ferrari	U
SERGIO	Sergio Schmitt	T

Navigation buttons: [Back] [Forward] [Search] [Add] [Delete] [Fechar]

Figura 5.3. A tela de manutenção da tabela de usuários.

5.3.1. Os Especialistas Técnicos de Informática

Os especialistas têm como papel principal, juntamente com os engenheiros do conhecimento, de alimentar e manter a base de casos. É o especialista que vai resolver problemas ainda não cobertos pelo sistema e fornecer a explicação da solução adotada para o engenheiro. Também cabe ao especialista perceber quando um caso da base está obsoleto, ou por ter uma solução diferente mais eficiente ou por algum avanço tecnológico que faça com que o problema não aconteça mais. Nesses casos, o caso deve ser corrigido ou apagado da base. Deve-se notar que sem um usuário do sistema fazendo o papel de especialista a base de casos se torna estática e pode ficar obsoleta rapidamente.

5.3.2. Os Engenheiros do Conhecimento

Os engenheiros do conhecimento são o elo de ligação entre os especialistas e o SE. Mais adiante veremos que a forma com que um caso é representado na base de casos pode ter um grau de abstração bastante elevado e o técnico nem sempre irá dominar a forma de transformar conhecimento real em um novo caso da base. Este profissional deve ter conhecimentos de informática e dominar a ciência da representação do conhecimento real na forma eletrônica.

5.3.3. Os Usuários Leigos

Os usuários leigos serão os maiores beneficiados do uso do sistema para descobrir defeitos em novos equipamentos. Quase sempre será ele que apresentará novos casos ao SE buscando uma solução dentre a base armazenada. Devemos sempre lembrar que o usuário leigo não faz manutenção na base de casos, acrescentando, removendo ou alterando casos. A ele cabe somente a consulta, e caso não seja encontrada uma solução satisfatória, um técnico deverá ser contatado e quando a solução real for encontrada por ele o engenheiro do conhecimento cria na base o novo caso.

5.4. A Representação de um Equipamento: a Arvore H/S

O início da interação do usuário com o sistema ao apresentar um novo caso, é justamente apontando qual é o equipamento defeituoso a ser consertado ou configurado. Para apontar ao SE qual dos equipamentos cobertos pelo domínio do conhecimento está sendo apresentado o usuário tem a disposição a árvore H/S.

Nome do campo	Tipo	Tamanho
COD_HS	Autoincrement	4
SINTETICO	Alpha	1
COD_ESTRUT	Alpha	5
DESCRICAO	Alpha	20

Tabela 5.2. A estrutura do arquivo de equipamentos.

Esta árvore está armazenada em um arquivo do sistema, onde cada registro representa um tipo de equipamento, e para cada equipamento são guardadas as seguintes informações: código reduzido, código estruturado, descrição do equipamento e tipo de representação sintética ou analítica⁴. A estrutura deste arquivo pode ser vista na tabela 5.2, a chave primária de acesso deste arquivo é o campo `cod_hs` e existe uma chave secundária de acesso através do campo `cod_estrut`.

5.4.1. A Árvore Hardware/Software: Organização e Indexação

Nem sempre um usuário leigo do sistema vai saber com precisão em que componente do equipamento está localizado o defeito. Assim o sistema deve prover uma forma de ser apontado o equipamento como um todo, e posteriormente o defeito será apontado dentre os casos filtrados. Por exemplo, imaginemos uma CPU com defeito no disco winchester. O usuário mais experiente poderia selecionar diretamente o registro da árvore H/S correspondente a defeitos de unidades de armazenamento, que é uma subdivisão do equipamento CPU. Porém o usuário leigo deve poder apontar somente CPU e receber como resposta, a seleção de todas palavras do dicionário referentes a CPU e dentre elas filtrar somente as que se referem ao seu problema.

Para facilitar a seleção e a escolha os registros da árvore H/S deverão ser apresentados aos usuários do sistema segundo a ordem de seu código estruturado, pois dessa forma ficará evidente a hierarquia presente nos componentes dos equipamentos. Por exemplo, o equipamento CPU tem código estruturado 01.00, significando que todos equipamentos que têm o código estruturado começando por “01.” são partes componentes do equipamento CPU. Na apresentação da árvore ao usuário visando a seleção do item com defeito vai aparecer então o código da CPU seguido de todos seus componentes, a código de monitor com seus componentes, e assim por diante.

5.4.2. Cadastrando um novo equipamento

Inicialmente, quando da definição do domínio do problema que este protótipo abrangeria, ficaram definidos que seriam seis os tipos de equipamentos com defeitos

⁴ Registros sintéticos são aqueles que representam grupos de componentes de um dado equipamento mas que não possuem casos vinculados diretamente a eles. Registros analíticos são vinculados a casos.

cobertos na base de casos: CPU's, impressoras, monitores, no-breaks, estabilizadores e scanners. Porém, independentemente deste fato, o sistema permite a inclusão de novos equipamentos a serem cobertos. Recomenda-se não expandir por demais o foco do SE sob a pena de haver perda de performance e aumento da dificuldade de busca numa base de casos muito grande.

A opção de inclusão de novos equipamentos no SE será mais comumente usada para o cadastro de uma nova subdivisão de um dos seis equipamentos já existentes. Por exemplo, os no-breaks poderiam ter subdivisões de acordo com suas partes componentes: circuito de entrada, baterias, transformador, inversor e circuito de saída. Enquanto o número de diferentes defeitos presentes em um dado equipamento for pequeno, a subdivisão só tende a complicar o encontro do caso com melhor *matching*, mas à medida que um equipamento tem muitos diferentes problemas, a divisão facilita o uso, como é o caso das CPU's, monitores e impressoras.

A tela de inclusão de novos equipamentos aparece abaixo na figura 5.4.

Código	Sintético	Cod. Estruturado	Descrição
22	S	03.00	Impressora
31	A	03.01	Matricial
32	A	03.02	Jato de Tinta
33	A	03.03	Laser
23	S	04.00	No-Break
24	S	05.00	Estabilizador
25	S	06.00	Scanner

Figura 5.4. A tela de inclusão de novos equipamentos.

5.4.3. Escolhendo um equipamento existente

Estando o novo equipamento com defeito cadastrado na árvore HS, o usuário deverá interagir com o sistema através do menu *Casos*, na opção *Apresentação de Novos Casos*, onde será feita a escolha do equipamento a partir da apresentação da árvore existente.

5.4.4. Quando não Sabemos o Equipamento Envolvido

O sistema especialista deve ser capaz de dar alguma resposta útil ao usuário mesmo quando este não tem certeza sobre onde está o defeito. Esta situação é mais comum do que podemos imaginar num primeiro momento. Alguns defeitos, segundo os técnicos especialistas responsáveis pela base de casos em uso neste protótipo, confundem o usuário e o levam a acreditar que o equipamento defeituoso é outro que não o verdadeiro.

Um exemplo muito claro é aquele onde não aparece imagem no monitor de vídeo quando ligamos o computador. Os usuários leigos tendem a imaginar que o problema está no monitor, quando na verdade, na maioria das vezes, o defeito está na CPU que não envia sinal de vídeo para o monitor. Na interação com o sistema o usuário vai procurar dentro dos defeitos de monitores aqueles que se parecem com o seu, mas o dicionário de PDD's não irá trazer nada muito parecido com o seu defeito particular.

Para estas situações o SE traz basicamente duas saídas diferentes: uma delas é permitir ao usuário selecionar mais de um equipamento na árvore, o que irá aumentando proporcionalmente o número de PDD's apresentadas ao usuário para que ele selecione as mais adequadas ao seu caso. A outra saída é o usuário selecionar um equipamento do tipo sintético, daqueles com o código estruturado terminando em “.00”, o que irá acarretar na apresentação de todas PDD's ligadas a todas divisões daquele equipamento.

5.5. A Representação dos Casos

A forma como representamos um caso num SE que utiliza raciocínio baseado em casos é obviamente um dos pontos-chave de funcionamento do sistema. Quando se consegue fazer uma abstração adequada da realidade, todos aspectos importantes de um dado objeto se fazem presentes de forma clara e lógica na representação digital daquele objeto dentro do programa.

No protótipo de SE aqui apresentado buscou-se fazer a representação dos casos da forma mais direta e simples possível, visando tanto facilitar o desenvolvimento do sistema, dado o limite de tempo para fazê-lo, quanto facilitar o uso por pessoas leigas. A idéia básica é representar o equipamento com defeito, a descrição do defeito e a descrição da solução. Adicionalmente acrescentou-se um código identificador de cada caso, a data de cadastramento do caso e a frequência com que o problema acontece⁵.

A estrutura do arquivo que armazena a representação de cada caso pode ser vista na tabela 5.3 e a chave primária de acesso deste arquivo é o campo `cod_caso`. Complementando esse arquivo ainda temos o dicionário de PDD's, descrito mais adiante, e o arquivo de equipamentos. Esses três arquivos compõem, como já foi dito anteriormente, a base de conhecimentos do sistema.

⁵ As possibilidades de frequência são: S=Sempre, I=Intermitente, R=Raro, N=Não sei.

Nome do campo	Tipo	Tamanho
COD_CASO	Autoincrement	4
COD_HS	Alpha	5
DESC_SUCINTA	Alpha	40
DESC_DETALHE	Alpha	200
SOLUCAO	Alpha	200
DATA_CAD_CASO	Date	8
FREQUENCIA	Alpha	1

Tabela 5.3. A estrutura do arquivo de casos.

5.5.1. Base de Casos: Organização e Indexação

O termo indexação, como vimos no capítulo 3, é empregado no raciocínio baseado em casos para designar técnicas utilizadas visando otimizar a busca na base de casos por um casamento adequado do caso novo com aqueles armazenados. Segundo a sua definição, os índices são combinações dos descritores importantes de um caso, isto é, daqueles descritores que distinguem um caso dos outros.

Os índices devem permitir que sejam reconhecidas as similaridades entre a situação corrente e os casos armazenados que podem contribuir para atingir os objetivos do caso corrente. Apesar do sucesso de muitos métodos automáticos de organização e indexação, descritos por WATSON (1994), Janet Kolodner alerta que os índices tendem a ser mais bem escolhidos pelo modo manual, e em aplicações práticas devem ser escolhidos desta forma.

No protótipo aqui apresentado os casos estão indexados primariamente de acordo com o equipamento em que o defeito foi observado. Desta forma, somente com a indicação do usuário de qual dos equipamentos apresentados está com problema já são descartadas da busca pelo algoritmo de recuperação da grande maioria dos casos, agilizando o casamento do melhor caso.

5.5.2. Cadastrando um Novo Caso

Quando o usuário do sistema procura como fazer para resolver um problema em equipamento coberto pelo domínio do conhecimento do SE e não encontra uma solução satisfatória, ele deve cadastrar no sistema seu caso deixando a solução em branco, para que seja preenchida posteriormente pelo engenheiro do conhecimento assim que um especialista descubra qual é a solução.

A decisão final sobre as soluções propostas pelo sistema serem válidas para o caso novo ou não é do usuário, mas o sistema sugere a atitude a ser tomada com base no grau de casamento calculado. Como será explicado mais adiante, se o índice de casamento ficar abaixo de 2 o sistema irá sugerir que o caso seja cadastrado como novo, e se o usuário concordar com a sugestão a tela apresentada na figura 5.4 será mostrada.

Ao usuário leigo não será liberado para preenchimento o campo solução, já que este somente pode ser cadastrado por especialistas ou engenheiros do conhecimento. Os seguintes campos já vêm preenchidos automaticamente pelo sistema: código do caso e data de cadastramento do caso.

Figura 5.5. A tela do programa de cadastro de novos casos

5.6. A Interação de um Novo Caso com o Sistema

Esta opção do sistema especialista deve ser a mais utilizada no dia-a-dia, já que o principal objetivo deste SE é encontrar soluções para problemas em equipamentos de informática, e é nesta parte do programa que este objetivo se concretiza. Esta apresentação de um novo caso foi dividida em três etapas, como num assistente.

O primeiro passo a ser realizado é a escolha na árvore HS indicando qual é o equipamento com defeito, se possível e aplicável deverá ser selecionado o subcomponente dos equipamentos (por exemplo no caso de CPU indicar se o problema está no ligamento, armazenamento, no windows, etc.) Se o equipamento não tiver divisões ou o usuário não souber em qual divisão está o problema, será escolhido o código de nível mais alto da árvore no que se refere àquele equipamento. Este primeiro passo pode ser visualizado na figura 5.5.

A escolha do equipamento irá disparar no sistema a montagem de um vetor contendo todas PDD's relacionadas. Como já foi dito no item 5.4.4, o usuário poderá escolher mais de um equipamento ou componente na dúvida de qual é o correto a ser apontado. As palavras do arquivo de dicionário de PDD's que, na opinião do usuário, são descritivas do problema que ele quer relatar serão marcadas na relação. Esta segunda etapa, de escolha das PDD's, pode ser visualizada na figura 5.6. Após esta etapa será iniciado o algoritmo de casamento, buscando os casos que mais se assemelham ao novo apresentado.

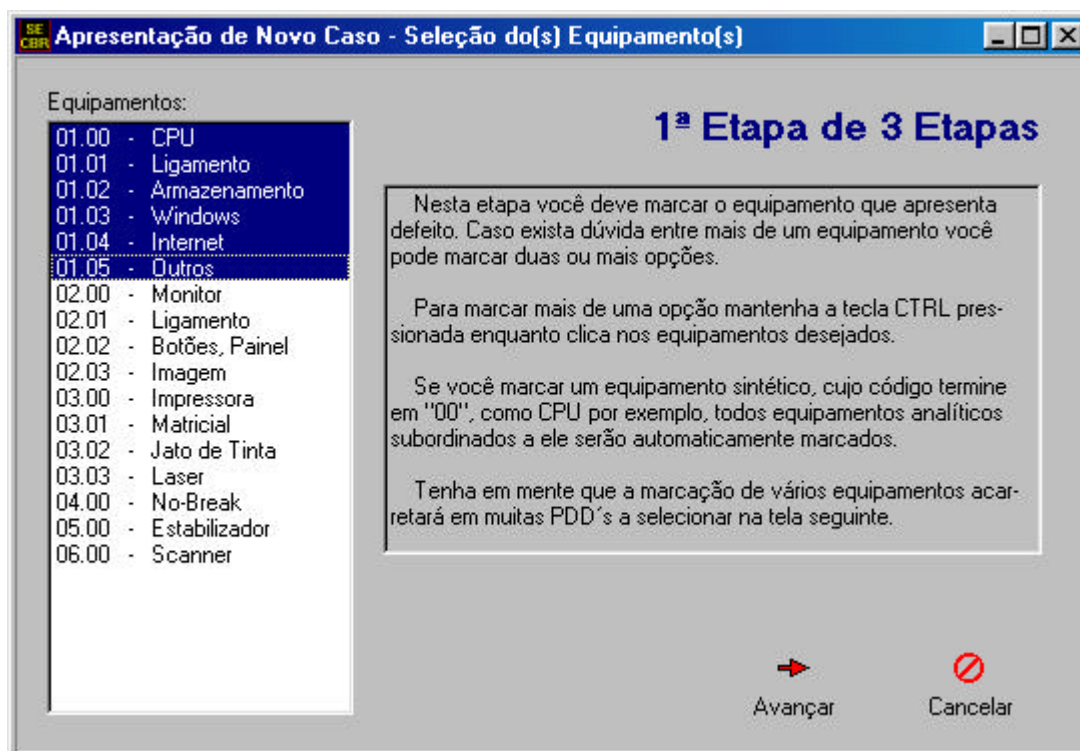


Figura 5.6. Selecionando o equipamento com defeito

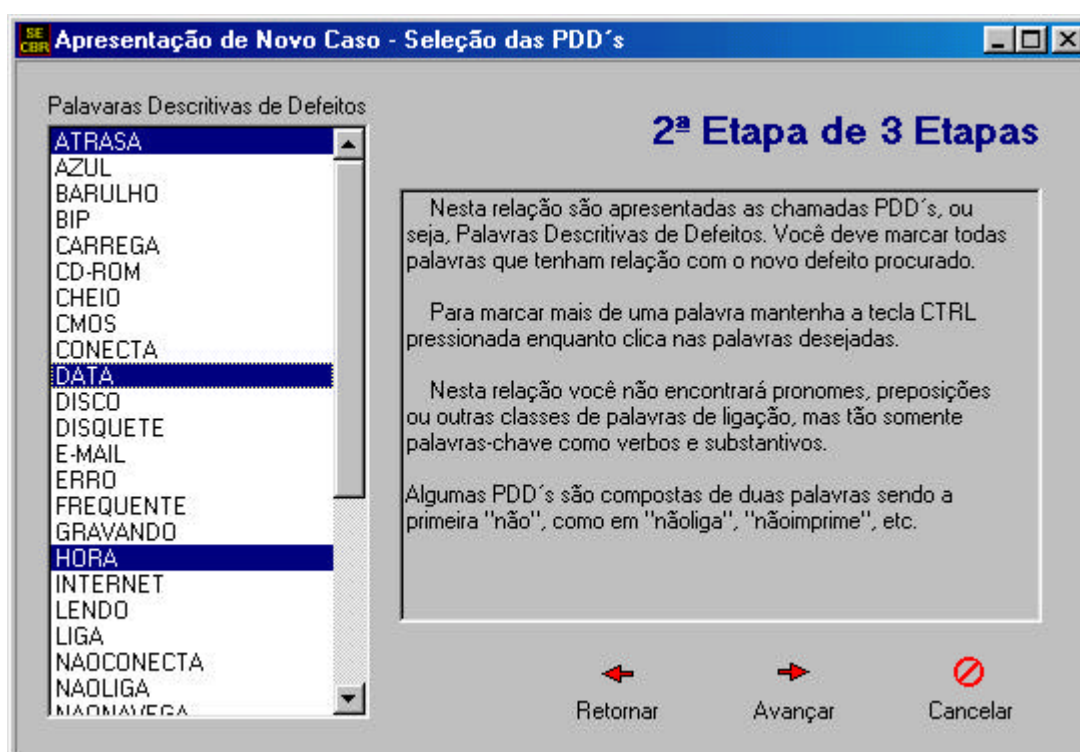


Figura 5.7. Selecionando Palavras Descritivas de Defeitos

SE CBR Apresentação de Novo Caso - Resultado

Caso: Código:

3ª Etapa de 3 Etapas

Descrição Sucinta:

Descrição Detalhada:

Solução:

Data Cadastro:

Frequência: ☒ Sempre ☐ Intermitente ☐ Raramente ☐ Não Sei

Grau:

Retornar Imprimir Atual Imprimir Possíveis Retornar Caso Avançar Caso Cancelar

Figura 5.8. Apresentação das possíveis soluções encontradas

Durante o processamento do algoritmo de casamento, todas PDD's do arquivo de dicionário que tem relação com o equipamento assinalado (ou os equipamentos quando for mais de um) serão varridas e quando uma que coincida com as palavras assinaladas pelo usuário como descritiva do defeito for encontrado o peso dela será acumulado à soma de pesos de seu caso. Ao final, teremos uma matriz com os casos que possuem PDD's iguais àquelas assinaladas pelo usuário, apresentada em ordem decrescente de peso acumulado, significando que o caso que mais se assemelha ao novo é o primeiro da lista. Se nenhum dos casos encontrados acumular 2 pontos ou mais é muito provável que este novo caso não tenha solução coberta pelo sistema, e será recomendado que seja cadastrado como novo caso, com a solução em branco. A figura 5.8 traz o algoritmo de casamento após a escolha das PDD's por parte do usuário.

5.6.1. A Apresentação das Possíveis Soluções

Depois de processado o algoritmo de casamento conforme vimos na figura 5.8, o sistema fará a apresentação da matriz de casos-resposta encontrada. Todo caso que tenha pelo menos uma PDD em comum com aquelas assinaladas pelo usuário será listado, mas aquele caso com maior probabilidade será o primeiro a ser mostrado, sendo sucedido em ordem decrescente de grau de probabilidade de solução. A figura 5.6. traz a tela do programa de interação de novos casos já com a matriz de casos-resposta sendo apresentada. Pode-se observar no campo da pontuação acumulada (identificado como grau) que este é o critério de classificação. Casos que obtiverem pontuação acumulada

abaixo de 2 apresentam semelhanças com o novo caso apresentado mas dificilmente será a solução procurada.

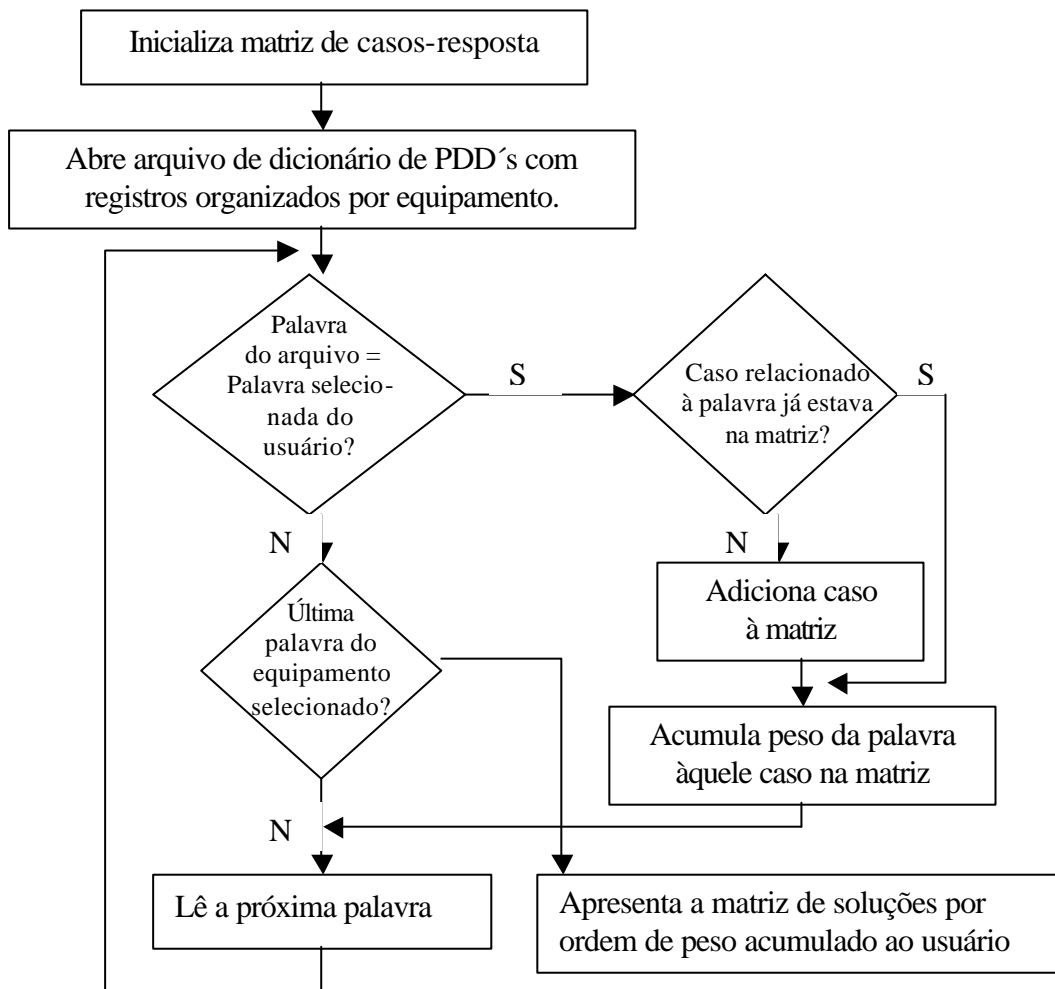


Figura 5.9. O fluxograma do algoritmo de casamento

5.6.2. Quando a Solução Proposta é Válida

O presente protótipo se propõe a ser um sistema de auxílio ao diagnóstico, pois a certeza do diagnóstico correto só poderá ser efetivamente dada pelo técnico que testar a solução no equipamento defeituoso. Do jeito como foi concebido, este sistema não armazena estatísticas sobre casos de diagnóstico bem-sucedidos (nem mal-sucedidos). Dessa forma, se a solução proposta é válida, ela simplesmente será implementada e nenhuma outra atitude será tomada no SE.

Caso a solução proposta pelo sistema não tenha sido exatamente a adotada mas levou à solução real por associação, o usuário deve convocar um especialista e um engenheiro do conhecimento para decidir se o caso armazenado na base de

conhecimentos pode ser alterado para abranger a nova situação, tornando-se mais generalista, ou se deve ser incluído um novo caso com a experiência recém adquirida.

5.6.3. Quando a Solução Proposta não é Válida

Um sistema especialista como este de nosso protótipo não irá sempre responder a todas questões apresentadas de maneira inequívoca. Faz parte da natureza dos sistemas desenvolvidos sob os paradigmas da IA a não necessidade de precisão nas respostas. Posto isso, veremos com frequência os usuários recebendo do sistema soluções propostas para resolver os defeitos em seus equipamentos que não são válidas, ou seja, não funcionam.

Nestes casos, durante a análise do sistema foi previsto que um novo registro de caso seria aberto onde apareceriam a descrição sucinta e a descrição detalhada do problema, com código estruturado do equipamento, código do caso e frequência, faltando somente descrever a solução. Um relatório de todos casos que tenham a solução em branco deverá ser emitido pelo usuário responsável pela manutenção do sistema, para que os técnicos especialistas busquem as soluções mais adequadas a cada caso e o engenheiro do conhecimento possa preencher adequadamente a base de conhecimentos.

5.7. Relatórios do Sistema

Dado que o sistema aqui apresentado é tão somente um protótipo, foram definidos inicialmente três tipos de relatórios de apoio aos usuários, sendo que os dois primeiros são emitidos ao final da apresentação de um novo caso ao sistema, utilizando os botões *Imprimir Atual* e *Imprimir Possíveis*, que podem ser vistos na figura 5.7, e o terceiro tem acesso via menu e consiste nos casos lançados no sistema e ainda sem solução. Qualquer um dos três relatórios é inicialmente visualizado em tela e o usuário escolhe posteriormente se deseja realmente imprimi-lo em papel ou não. Estes relatórios estão explicados logo a seguir.

5.7.1. Listagem das Soluções Possíveis em um Novo Caso

Ao final da apresentação de um novo caso ao sistema a tela anteriormente vista na figura 5.7 traz dois botões de listagem: *Imprimir Atual* e *Imprimir Possíveis*. O primeiro botão resultará num relatório de uma página com os dados relativos ao caso atualmente apresentado na tela e servirá normalmente para o usuário que tem o equipamento a ser consertado longe do micro com o SE e que deseja levar a solução impressa para aplica-la. O segundo botão trará um relatório com um número de páginas igual ao número de soluções encontradas, mostrando um caso por página. Este relatório possibilita ao usuário ir testando uma solução por vez quando existe dúvida entre as várias soluções apresentadas pelo sistema.

Além disso, o botão imprimir possíveis traz uma funcionalidade que não fica clara à primeira vista, que é a de listar todos os casos relativos a um determinado equipamento. Se o usuário marcar todas as PDD's mostradas na etapa 2, serão apresentados na etapa 3 todos os casos relativos ao equipamento marcado na primeira etapa. Um usuário leigo pode usar este relatório quando o resultado de uma busca sua for uma longa lista de casos possíveis e este quiser analisar as possibilidades com calma tendo-as impressas no papel. O Engenheiro do Conhecimento poderá usar uma relação de casos de um determinado equipamento para dar manutenção na base de conhecimento, eliminando aqueles casos obsoletos ou estudando a possibilidade de dividir um equipamento em suas partes componentes quando o número de casos daquele código estruturado fica muito grande e a busca ineficiente.

SERBC			
Auxílio e Diagnóstico de Defeitos em Computadores			Data : 29/09/02 Hora : 18:14
Caso : 20	Código : 01.05	Data Cadastro : 31/07/02	Frequência : S
Descrição Sucinta : Atrasa data e/ou hora			
Descrição Detalhada : Atrasa data e/ou hora			
Solução :			
Verificar voltagem da bateria CMOS. Se estiver baixa trocar. Caso contrário é problema na placa-mãe.			

Figura 5.10. Listagem das soluções possíveis em um novo caso

5.7.2. Relação de casos ainda sem solução

Com certeza para o usuário leigo que busca respostas no SE, uma relação de problemas sem solução não serve para nada, mas para o especialista e o engenheiro do conhecimento é um relatório fundamental. É através dessa relação que o sistema pode crescer, incorporando novos conhecimentos a sua base de casos à medida que os casos não resolvidos vão sendo preenchidos. Como já vimos anteriormente, um caso fica cadastrado no sistema com a solução em branco quando um novo caso é apresentado por um usuário e nenhuma das respostas encontradas satisfaz. A figura 5.10. apresenta o leiaute deste relatório.

SERBC		Data : 29/09/02	
Auxílio e Diagnóstico de Defeitos em Computadores		Hora : 18:13	
<hr/>			
Listagem de Casos sem Solução			
<hr/>			
Caso : 34	Código : 02.03	Data Cadastro : 31/07/02	Frequência : S
Descrição Sucinta : Demora para vir imagem quando liga			
Descrição Detalhada : Demora para aparecer a imagem depois de ligar			

Figura 5.11. Listagem dos casos ainda sem solução

5.8. Avaliando o Protótipo

Uma vez que o protótipo esteja desenvolvido e funcional, resta saber se o mesmo é eficiente, ou melhor, medir o grau de eficiência do sistema a fim de saber se ele pode realmente auxiliar no diagnóstico de problemas em computadores. Para isso, deve-se desenvolver uma forma de avaliação do SE, em situação real de trabalho e verificar se suas respostas são as mesmas dos especialistas humanos.

Como vimos no capítulo 4, a base de casos foi formada a partir de um banco de dados de mais de 10.000 ordens de serviço preenchidas no período de setembro de 1998 a julho de 2001. O mês de julho de 2001 marcou o início dos trabalhos de análise do banco de dados e transformação deste em base de casos, mas os especialistas da empresa onde as ordens foram coletadas continuaram trabalhando e preenchendo novas ordens, a uma razão média de pouco mais de 300 novas OS por mês, resultando em mais de 4.000 ordens inéditas até a conclusão do protótipo.

Estas novas OS, inéditas para o sistema, formam a base dos testes de avaliação do protótipo aqui desenvolvido e apresentado, da OS de número 10.805 a OS de número 15.000. De maneira aleatória foram escolhidas 100 ordens de serviço inéditas e o defeito nelas relatado foi apresentado ao sistema e a solução sugerida pelo mesmo foi comparada com a solução preenchida pelo técnico especialista.

Para a escolha aleatória das ordens foi usada a função =ALEATÓRIO() do Microsoft Excel, que retorna um número em ponto flutuante entre 0 e 1. Foram geradas 350 células com a seguinte fórmula:

=INT(ALEATÓRIO()*4195+10805)

o que garante números randômicos no intervalo entre 10.805 e 15.000, que é o intervalo dos números das ordens inéditas a serem usadas nos testes. Foram necessários 334 números aleatórios diferentes para obter 100 O.S. válidas, um aproveitamento de cerca

de 29,94%, quando na primeira fase 10.804 ordens resultaram em 3.966 válidas, tendo um aproveitamento de 36,71%. Como relatamos no capítulo 4, algumas ordens de serviço não são relacionadas a defeitos reais, ou não possuem a solução preenchida, ou ainda se referem a equipamentos não cobertos pelo domínio do conhecimento deste protótipo. Sendo assim, cada número de OS sorteado era conferido e filtrado a fim de obter-se 100 ordens aproveitáveis, e encima somente de ordens aproveitáveis que a análise foi feita.

Para interagir com o sistema escolheu-se uma pessoa com conhecimento intermediário de informática, um estudante do 4º semestre do curso de informática da UNIJUÍ. Essa informação é relevante pois durante a apresentação dos casos ao sistema, o usuário deve escolher quais palavras mostradas melhor descrevem o defeito do novo caso, e esta escolha envolve um grau de subjetividade, que pode resultar em respostas diferentes com diferentes usuários. Na busca de obter um comportamento médio do sistema buscou-se um usuário que não fosse especialista mas também que não fosse totalmente leigo.

Para uma avaliação mais acurada deveriam ser feitos testes com pessoas com nível de conhecimentos diferentes acerca do domínio do problema coberto pelo sistema e os resultados comparados entre si, mas o tempo restante no cronograma para a avaliação do sistema não permitiu tal trabalho. Isso fica como sugestão para trabalhos futuros como descrito no capítulo 6.

O resultado da avaliação do protótipo aparece na tabela 5.4, e como podemos perceber pela sua análise, o sistema apresenta um grau bastante alto de confiabilidade nas suas respostas.

	Grupo 01	Grupo 02	Grupo 03	Grupo 04	Grupo 05	Grupo 06	TOTAL
OS de equipamentos com defeito	54	10	31	4	1	0	100
Diagnósticos corretos do SE	34	7	24	1	1	0	67
Casos não cobertos pelo SE	12	2	2	3	0	0	19
Diagnósticos errôneos do SE	8	1	5	0	0	0	14
Percentual de acerto por grupo	63%	70%	77%	25%	100%	-	67%

Tabela 5.4. Resultado da avaliação do protótipo SADPC

A tabela acima retrata uma avaliação mais quantitativa do que qualitativa, mostrando basicamente números, mas pode ser bastante esclarecedora se devidamente abordada. Os casos onde o sistema falhou foram divididos para melhor análise em casos não cobertos pelo SE e diagnósticos errôneos do SE.

Os casos não cobertos pelo SE são aqueles onde as PDD's apresentadas ao usuário não se encaixavam no seu novo caso e por consequência a base de conhecimento atual do sistema não tem uma solução satisfatória em um caso semelhante para ser apresentado. Esses casos não devem ser encarados como um fracasso do

sistema, pois uma base de conhecimento nunca será capaz de cobrir todas as situações reais do dia-a-dia. Uma das características que fazem de um SE um software inteligente é a sua capacidade de aprender. Esses casos não cobertos pelo sistema seriam cadastrados como novos casos, onde o índice acumulado de pontos não atingiu o ponto mínimo (soma acumulada de 2 pontos), e ficariam com o campo da solução em branco aguardando ser preenchida pelos especialistas em conjunto com o engenheiro do conhecimento. É uma situação normal de ocorrer, e se for considerada assim a eficiência média do sistema na avaliação passa de 67% para 86%.

Já os diagnósticos errôneos do sistema são aqueles casos onde as PDD's selecionadas pelos usuários fizeram com que o caso mais provável apresentado não tivesse a solução que de fato resolveria o novo caso apresentado. Ou seja, a solução sugerida pelo SE não resolveria o novo caso. Estes casos representaram 14% dos 100 casos selecionados para avaliação e podemos considerá-los como falhas do sistema. Essas falhas podem ser corrigidas analisando-se os casos em que isso ocorre e buscando PDD's que diferenciem os casos já armazenados na base de conhecimento destes novos apresentados e cadastrando os novos casos com suas próprias PDD's e com a solução em branco. Uma possibilidade que não foi pensada no protótipo aqui apresentado e que poderia vir a melhorá-lo seria a possibilidade de cadastrar os casos diagnosticados erroneamente para que os especialistas e o engenheiro do conhecimento achem as soluções corretas e as PDD's que melhor descrevem o caso diferenciando-o daquele já armazenado.

Os números da análise separados por grupo ainda trazem algumas peculiaridades. Podemos perceber que os primeiros 3 grupos, além de concentrarem a grande maioria dos casos (95% dos casos) ainda mantém, cada um deles, a média de acertos próxima da média geral. Os últimos 3 grupos destoam muito da média, sendo que o último, o dos scanners, nem apresentou caso sorteado. O grupo 04, dos no-breaks teve 3 dos 4 casos sorteados não cobertos pelo sistema, o que denota uma grande variedade de problemas possíveis, ou pelo menos uma falta de PDD's adequadas para descrever os problemas apresentados.

6. Conclusões e Considerações Finais

6.1. Conclusões

O presente trabalho se propôs a estudar a aplicabilidade de Sistemas Especialistas com Raciocínio Baseados em Casos como ferramenta de suporte à solução de problemas surgidos em microcomputadores, conforme descrito em seu objetivo geral na introdução. Para isso foi feita a análise e a modelagem de um sistema especialista, desenvolvido utilizando o paradigma do raciocínio baseado em casos aplicado sobre um banco de dados pré-existente de 10.804 ordens de serviço de uma empresa de assistência técnica.

A aplicabilidade pode ser vista sob dois pontos de vista, que são: em primeiro lugar ser *possível* modelar o tipo de problema aqui apresentado com o paradigma RBC, e em segundo lugar ser *viável* o uso de um sistema de auxílio para diagnóstico de problemas em computadores. A questão da possibilidade foi comprovada a partir da elaboração do formato que a base de conhecimentos iria ter e da apresentação do algoritmo de casamento que recuperaria os casos dessa base. Já a viabilidade foi comprovada durante a avaliação do protótipo, aonde vimos que 67% dos casos apresentados ao SE foram corretamente diagnosticados e mais 19% dos casos poderiam ter sido corretamente diagnosticados mas não estavam presentes na base de casos, podendo então chegar a eficiência do sistema a 86%. Podemos concluir então pelos números apresentados que o objetivo geral do trabalho foi plenamente atingido.

Além do objetivo geral, foram traçados 4 objetivos específicos, que passaremos a analisar. O primeiro destes objetivos, que consistia na disseminação do estudo e uso do RBC, já foi atingido. Este trabalho já rendeu 2 palestras para um total de mais de 100 alunos de informática em Três de Maio e Horizontina e a própria dissertação auxilia nesta disseminação. O segundo objetivo era avaliar o atual estado da arte do RBC, que pôde ser concretizado com as pesquisas bibliográficas feitas para a escrita do segundo e terceiro capítulos, especialmente com a leitura dos artigos da 4ª Conferência Internacional em Raciocínio Baseado em Casos realizada em Vancouver, em 2001. O terceiro objetivo consistia em propor novas funcionalidades em SE's que usam RBC específicas para o domínio do conhecimento estudado. Acreditamos que o conceito adotado na máquina de inferência do protótipo apresentado, de palavras descritivas de defeitos, tenha trazido uma possibilidade a mais no desenvolvimento de SE's que usam RBC, já que permite uma forma diferente de interação com o usuário. O quarto objetivo era desenvolver um protótipo funcional de um sistema computacional que demonstre o RBC em ação, o que foi feito e demonstrado no capítulo 5.

6.2. Recomendações e Considerações Finais

É normal chegar ao final do desenvolvimento de um trabalho extenso, do porte deste aqui apresentado, com a sensação de que mais coisas poderiam ter sido feitas, certos tópicos poderiam ter sido mais bem desenvolvidos ou exemplificados, apesar da exaustiva revisão feita. É como o desenvolvimento de um software: sempre é possível

apresentar uma nova versão com melhorias e implementação de sugestões. Mas com certeza este trabalho não termina aqui. Este é somente o final de uma etapa e neste tópico serão apresentadas algumas recomendações e sugestões para o futuro.

A primeira questão a ser levantada é a forma de interação do usuário com o sistema. Sabe-se que a maneira mais adequada seria possibilitar a descrição do defeito no equipamento em linguagem natural, como: *a imagem do monitor oscila*, e esse não foi o jeito adotado em função do tempo reduzido para o desenvolvimento do protótipo. Teria de ser construído um analisador sintático e semântico, auxiliado por um dicionário de sinônimos, para que o sistema entendesse, por exemplo, que vídeo, tela e monitor se tratam do mesmo equipamento. O arquivo de usuário, o banco de dados de casos e a árvore de equipamentos poderiam continuar os mesmos. O dicionário de PDD's que seria substituído por outro mecanismo, constituindo uma nova máquina de inferência, e um novo algoritmo de casamento deveria ser construído.

Após a realização da avaliação do protótipo, onde foram constatados 14 casos diagnosticados erroneamente pelo sistema, de um total de 100 casos, o autor levantou outra questão a ser desenvolvida: a possibilidade de se criar no protótipo uma funcionalidade extra, que permitisse o cadastro destes casos. Dessa forma os especialistas e o engenheiro do conhecimento poderiam buscar as soluções corretas e as PDD's que melhor descrevem este novo caso, de forma a diferenciá-lo daquele já armazenado. Talvez o próprio arquivo de casos possa ser usado, com o acréscimo de um campo de controle que marcasse esses casos como incompletos enquanto não solucionados, mas diferenciando-os daqueles não cobertos pelo sistema, que estarão simplesmente cadastrados com a solução em branco.

Os testes de avaliação do protótipo também comprovaram uma realidade já detectada na fase de filtragem do banco de dados de ordens de serviço, descrita no capítulo 4: os equipamentos cujo código estruturado começa por 05 e 06 – estabilizadores e scanners, têm uma representatividade bastante pequena quando comparados com os quatro primeiros itens – CPU's, monitores, impressoras e no-breaks. Das 100 ordens de serviço sorteadas para a avaliação, somente uma se referia ao grupo 05, dos estabilizadores, e nenhuma ao grupo 06, dos scanners. Do total de ordens de serviço que compuseram a base de casos, 0,96% se referiam ao grupo 05 e 0,47% ao grupo 06, como pode ser constatado na tabela 4.10. Esses números levam à conclusão que estes 2 grupos poderiam ser eliminados do sistema com pouca perda para os usuários e com o ganho de maior foco, o que sempre resulta em maior eficiência de um SE.

Finalmente a última recomendação para futuros trabalhos é a adoção de um número maior de pessoas fazendo os testes de avaliação do protótipo, com nível de conhecimentos diferentes para se obter um nível de confiança maior nos resultados apresentados.

Anexo 1. A Árvore de Equipamentos do Sistema

COD_HS	COD_ESTRUT	SINTETICO	DESCRIÇÃO
1	01.00	S	CPU
2	02.00	S	Monitor
3	03.00	S	Impressora
4	04.00	S	No-Break
5	05.00	S	Estabilizador
6	06.00	S	Scanner
7	01.01	A	Ligamento
8	01.02	A	Armazenamento
9	01.03	A	Windows
10	01.04	A	Internet
11	01.05	A	Outros
12	03.01	A	Matricial
13	03.02	A	Jato de Tinta
14	03.03	A	Laser
15	02.01	A	Ligamento
16	02.02	A	Botões, Painei
17	02.03	A	Imagem

Anexo 2. O Arquivo de PDD's do Sistema

COD_ESTRUT	COD_CASO	PALAVRA	GRAU
01.01	1	Naoliga	2
01.01	2	Liga	1
01.01	2	Naosinal	2
01.01	3	Bip	2
01.01	3	Naosinal	2
01.01	4	Carrega	1
01.01	4	Erro	1
01.01	4	Naowindows	2
01.01	4	Winchester	2
01.01	5	Carrega	1
01.01	5	Cmos	2
01.01	5	Erro	1
01.01	5	Naowindows	2
01.01	6	Naowindows	2
01.02	7	Disquete	2
01.02	7	Erro	1
01.02	8	Erro	1
01.02	8	Winchester	2
01.02	9	Cd-Rom	2
01.02	9	Erro	1
01.02	9	Lendo	2
01.02	10	Cheio	2
01.02	10	Disco	2
01.02	10	Erro	1
01.02	11	Erro	1
01.02	11	Gravando	2
01.02	11	Lendo	2
01.03	12	Erro	1
01.03	12	Registro	2
01.03	13	Azul	2
01.03	13	Erro	1
01.03	13	Protecao	2
01.03	13	Tela	1
01.04	15	Internet	1
01.04	15	Naoconecta	2
01.04	16	Conecta	1
01.04	16	Internet	1
01.04	16	Naonavega	2
01.04	17	E-Mail	2
01.04	17	Naorecebe	2
01.04	18	E-Mail	2
01.04	18	Naotransmite	2
01.05	19	Barulho	2

01.05	20	Atrasa	2
01.05	20	Data	2
01.05	20	Hora	2
01.05	21	Erro	2
01.05	21	Programa	1
01.05	21	Sempre	1
01.05	21	Trava	2
01.05	22	Erro	2
01.05	22	Frequente	1
01.05	22	Programa	1
01.05	22	Trava	2
02.01	23	Naoliga	2
02.01	24	Naoimagem	2
02.02	25	Botao	2
02.02	25	Naoliga	1
02.02	25	Naosegura	1
02.03	26	Liga	1
02.03	26	Naoimagem	2
02.03	27	Imagem	1
02.03	27	Oscila	2
02.03	28	Alteradas	1
02.03	28	Cores	2
02.03	29	Cabo	2
02.03	29	Cores	1
02.03	29	Mexer	1
02.03	30	Faixa	2
02.03	30	Naoimagem	1
02.03	30	Vertical	1
02.03	31	Faixa	2
02.03	31	Horizontal	1
02.03	31	Naoimagem	1
02.03	32	Foco	2
02.03	32	Imagem	1
02.03	33	Brilho	2
02.03	33	Fraca	1
02.03	33	Imagem	1
02.03	34	Demora	2
02.03	34	Imagem	1
02.03	35	Escura	2
02.03	35	Imagem	1
03.01	36	Naoliga	2
03.01	37	Naoimprime	2
03.01	37	Naosinal	1
03.01	38	Luzes	2
03.01	38	Naoimprime	2
03.01	38	Piscam	1

03.01	39	Carro	1
03.01	39	Tranca	2
03.01	40	Alinhamento	2
03.01	41	Falhas	2
03.01	41	Horizontais	1
03.01	41	Linhas	2
03.01	42	Naopuxa	2
03.01	42	Papel	1
03.01	43	Barulho	2
03.01	44	Fita	2
03.01	44	Naotraciona	2
03.01	45	Botao	2
03.01	45	Naofunciona	1
03.01	45	Painel	2
03.01	46	Borra	2
03.01	46	Impressao	1
03.01	47	Folha	1
03.01	47	Naoimprime	1
03.01	47	Pula	2
03.01	48	Erro	1
03.01	48	Falta	2
03.01	48	Papel	2
03.01	49	Papel	1
03.01	49	Pula	1
03.02	50	Naoliga	2
03.02	51	Naoimprime	2
03.02	51	Naosinal	1
03.02	52	Luzes	2
03.02	52	Naoimprime	2
03.02	52	Piscam	2
03.02	53	Naoimprime	2
03.02	53	Papel	1
03.02	53	Puxa	2
03.02	54	Barulho	2
03.02	55	Borra	2
03.02	55	Impressao	1
03.02	56	Caracteres	2
03.02	56	Estranhos	1
03.02	57	Cartuchos	2
03.02	57	Naoreconhece	1
03.02	58	Alinhamento	2
03.02	59	Naocolorido	2
03.02	60	Naopreto	2
03.02	61	Algumas	1
03.02	61	Naocores	2
03.02	62	Falhas	2

03.02	62	Linhas	1
03.02	63	Copias	1
03.02	63	Erro	1
03.02	63	Muitas	1
03.03	64	Naoliga	2
03.03	65	Naoimprime	2
03.03	65	Naosinal	1
03.03	66	Luzes	2
03.03	66	Naoimprime	2
03.03	66	Piscam	2
03.03	67	Naopuxa	2
03.03	67	Papel	1
03.03	68	Barulho	2
03.03	69	Borra	2
03.03	69	Impressao	1
03.03	70	Caracteres	1
03.03	70	Estranhos	1
03.03	71	Falhas	2
03.03	71	Linhas	1
03.03	72	Atola	2
03.03	72	Papel	2
04.00	73	Desliga	1
04.00	73	Falta	1
04.00	73	Luz	1
04.00	73	Segura	2
04.00	74	Naoliga	2
04.00	75	Bateria	1
04.00	75	Bip	2
04.00	75	Inversor	1
05.00	76	Naoliga	2
05.00	77	Barulho	2
05.00	77	Vibricao	1
05.00	78	Barulho	2
05.00	78	Cliques	1
05.00	79	Cheiro	2
06.00	80	Lâmpada	1
06.00	80	Naoliga	2
06.00	81	Naocomunica	2
06.00	82	Adquire	1
06.00	82	Naoescaneia	2
06.00	83	Lampada	2
06.00	83	Naoacende	2
06.00	84	Escaneia	1
06.00	84	Trava	2

Anexo 3. O Arquivo de Casos do Sistema

Caso	Cod_hs	Desc_sucinta/Desc_detalhe/Solucao	Data_cad	Freq
1	01.01	Não Liga	31/07/02	S
		CPU não liga e não acende nenhum led nem ventoinhas.		
		Problema na alimentação de energia. Trocar fonte.		
2	01.01	Liga mas não dá nenhum sinal	31/07/02	S
		CPU liga leds e ventoinhas mas não tem sinal de vídeo.		
		Problema na memória. Pode ser placa de vídeo ou placa mãe com menor probabilidade.		
3	01.01	Liga mas só dá bip sem sinal de vídeo	31/07/02	S
		CPU liga leds e ventoinhas, dá bip e não tem sinal de vídeo		
		Problema na memória ou placa de vídeo. Testar primeiro trocando uma depois outra.		
4	01.01	Não carrega Windows, erro HDD	31/07/02	S
		CPU liga mas não carrega Windows, acusando erro no HDD		
		Testar possibilidade de mau contato. Caso não seja trocar HD.		
5	01.01	Não carrega Windows, erro CMOS	31/07/02	S
		CPU liga mas não carrega Windows, acusando erro na CMOS		
		Verificar voltagem da bateria CMOS. Se estiver baixa trocar. Caso contrário é problema na placa-mãe.		
6	01.01	Não carrega Windows	31/07/02	S
		CPU liga mas não carrega Windows		
		Arquivos corrompidos ou faltando. Copiar os arquivos ou reinstalar o SO.		
7	01.02	Erro lendo/gravando disquete	31/07/02	I
		Acusa erro durante a leitura ou gravação de disquetes		
		Tentar com outros disquetes. Continuando o erro trocar drive.		
8	01.02	Erro lendo/gravando winchester	31/07/02	I
		Acusa erro durante a leitura ou gravação do winchester		
		Passar scandisk. Caso volte a acontecer trocar o HD.		
9	01.02	Erro lendo CD	31/07/02	I
		Acusa erro durante a leitura de CDs		
		Fazer limpeza no conjunto ótico do CD.		
10	01.02	Erro de disco cheio	31/07/02	I
		Acusa erro de disco cheio		
		Apagar arquivos temporários, lixeira e arquivo inúteis.		
11	01.02	Erro em outras mídias	31/07/02	N
		Acusa erro durante o uso de outras mídias		
		Testar com outras mídias. Persistindo fazer manutenção e limpeza da unidade.		
12	01.03	Erro de registro	31/07/02	I
		Acusa erro no registro do Windows		
		Passar programa reparador de registro como o scanreg. Testar memórias.		
13	01.03	Erro geral de proteção. Tela azul.	31/07/02	I
		Acusa erro geral de proteção (General Protect Failure). Dá tela azul.		

		Trocar memória RAM ou desabilitar cache externa. Pesquisar programas recentemente instalados. Passar antivírus atualizado.		
14	01.04	Não disca para internet	31/07/02	S
		Não faz a discagem para o provedor internet		
		Trocar placa de modem e testar sinal de linha com aparelho telefônico.		
15	01.04	Não conecta internet	31/07/02	N
		Disca mas não conecta no provedor internet		
		Testar em outra linha e com outro provedor. Verificar vírus. Persistindo trocar o modem.		
16	01.04	Conecta mas não navega	31/07/02	S
		Conecta no provedor mas não abre nenhuma página		
		Verificar os arquivos responsáveis pela navegação, devem estar corrompidos. Passar anti-vírus.		
17	01.04	Conecta mas não recebe e-mails	31/07/02	S
		Conecta no provedor mas não recebe e-mails		
		Verificar configurações do servidor POP3 no programa usado para e-mails.		
18	01.04	Conecta mas não envia e-mails	31/07/02	N
		Conecta no provedor mas não envia e-mails		
		Verificar configurações do servidor SMTP no programa usado para e-mails.		
19	01.05	Faz barulho estranho	31/07/02	I
		CPU faz barulho estranho		
		Verificar ventoinhas do processador e fonte. Com menor probabilidade pode ser o HD com defeito.		
20	01.05	Atrasa data e/ou hora	31/07/02	S
		Atrasa data e/ou hora		
		Verificar voltagem da bateria CMOS. Se estiver baixa trocar. Caso contrário é problema na placa-mãe.		
21	01.05	Trava ou dá erro em qualquer programa	31/07/02	I
		CPU trava ou dá mensagem de erro usando qualquer programa		
		Trocar memória RAM ou desabilitar cache externa. Pesquisar programas recentemente instalados. Verificar ventoinhas.		
22	01.05	Trava ou dá erro em programa específico	31/07/02	I
		CPU trava ou dá mensagem de erro usando um programa específico		
		Desinstalar e reinstalar o programa. Se não resolver reinstalar também o SO.		
23	02.01	Monitor não liga	31/07/02	S
		Monitor não liga		
		Problema de alimentação de energia. Trocar a fonte ou os componentes queimados da mesma.		
24	02.01	Monitor liga mas não tem imagem	31/07/02	S
		Monitor liga mas não tem imagem		
		Verificar circuitos de deflexão horizontal e vertical.		
25	02.02	Botão de ligar não segura ligado	31/07/02	I
		Botão liga/desliga não segura ligado		
		Trocar o botão		
26	02.03	Imagem do monitor oscila/treme	31/07/02	I
		Imagem do monitor oscila/treme		
		Trocar o potenciômetro que regula a posição horizontal e/ou vertical da		

		imagem.		
27	02.03	Cores aparecem alteradas	31/07/02	N
		Cores aparecem alteradas		
		Solda fria na placa CRT.		
28	02.03	Cores alteradas mexendo no cabo	31/07/02	I
		Cores alteram quando se mexe no cabo de sinal de vídeo		
		Fio rompido no cabo de sinal de vídeo. Cortar e ressoldar.		
29	02.03	Só aparece faixa vertical de luz	31/07/02	N
		Só aparece faixa vertical de luz no meio		
		Defeito em componente da deflexão horizontal. Aconselhável esquema eletrônico para efetuar conserto.		
30	02.03	Só aparece faixa horizontal de luz	31/07/02	N
		Só aparece faixa horizontal de luz no meio		
		Defeito em componente da deflexão vertical. Aconselhável esquema eletrônico para efetuar conserto.		
31	02.03	Imagem fora de foco	31/07/02	S
		Imagem fora de foco		
		Regular botão de foco no fly-back. Se o problema oscila de com o tempo do monitor ligado e não fica 100% bom tem que trocar o fly-back.		
32	02.03	Imagem clara demais, muito brilho	31/07/02	S
		Imagem clara demais, muito brilho		
		Regular botão no fly-back. Se não ficar 100% bom tem que trocar o fly-back.		
33	02.03	Demora para vir imagem quando liga	31/07/02	S
		Demora para aparecer a imagem depois de ligar		
34	02.03	Imagem muito escura	31/07/02	S
		Imagem muito escura		
35	03.01	Não liga	31/07/02	S
		Impressora não liga		
		Problema de alimentação de energia. Trocar a fonte ou os componentes queimados da mesma.		
36	03.01	Não imprime e não dá sinal nenhum	31/07/02	S
		Liga mas não imprime e não dá nenhum sinal		
		Impressora não está recebendo dados. Provável defeito na interface de comunicação da impressora ou do micro. Testar em outro micro para isolar o defeito.		
37	03.01	Não imprime e piscam luzes do painel	31/07/02	S
		Não imprime e piscam luzes do painel		
		Verificar no manual da impressora o significado das luzes. Normalmente é sinal que o carro de impressão está trancado.		
38	03.01	Tranca o carro durante a impressão	31/07/02	I
		Tranca o carro durante a impressão		
		Fazer manutenção na impressora. Limpeza e lubrificação do eixo do carro de impressão.		
39	03.01	Imprime fora do alinhamento	31/07/02	S

Imprime caracteres fora do alinhamento vertical				
Folga da cabeça de impressão no eixo. Trocar as buchas ou ajustar conforme necessário.				
40	03.01	Imprime com linhas horizontais falhadas	31/07/02	S
Imprime com linhas horizontais falhadas cortando os caracteres Falhando uma ou mais agulhas da cabeça de impressão.				
Pode ser agulha gasta ou a bobina que dispara a agulha queimada. Fazer recondicionamento da cabeça de impressão ou troca.				
41	03.01	Não puxa papel	31/07/02	N
Não puxa papel				
Fazer limpeza, lubrificação e revisão geral.				
42	03.01	Barulho estranho ao imprimir	31/07/02	I
Barulho estranho ao imprimir				
Fazer manutenção na impressora. Limpeza e lubrificação do eixo do carro de impressão. Verificar buchas do conjunto guia da cabeça.				
43	03.01	Não traciona a fita	31/07/02	S
Não traciona a fita				
Engrenagem do conjunto tracionador gasta ou quebrada. Substituir.				
44	03.01	Não funciona botão do painel	31/07/02	I
Não funciona botão do painel				
Remover painel e verificar conexões. Pode ser mau contato, cabo flat quebrado ou oxidado ou botão quebrado.				
45	03.01	Borra a impressão	31/07/02	I
Borra a impressão com manchas horizontais				
A cabeça de impressão está muito próxima do papel. Afastar usando alavanca de posicionamento da cabeça. Se a impressão ficar muito fraca deve-se fazer recondicionamento da cabeça de impressão.				
46	03.01	Pula folhas sem imprimir	31/07/02	I
Pula folhas sem imprimir				
Verificar driver de impressão usado. Caso esteja correto pode ser defeito na interface de comunicação da impressora.				
47	03.01	Acusa falta de papel mesmo tendo	31/07/02	S
Acusa falta de papel mesmo quando tem				
Sensor de indicação de presença de papel com problema. Trocar ou fechar curto para desabilitá-lo.				
48	03.01	Pula página além do que devia	31/07/02	I
Pula o picote da página além do que devia				
Configuração incorreta do tamanho da página em uso. Verificar na configuração da eeprom da impressora e no software em uso.				
49	03.02	Não liga	31/07/02	S
Não liga				
Problema de alimentação de energia. Trocar a fonte ou os componentes queimados da mesma.				
50	03.02	Não imprime e não dá sinal nenhum	31/07/02	S
Não imprime e não dá nenhum sinal				
Impressora não está recebendo dados. Provável defeito na interface de comunicação da impressora ou do micro. Testar em outro micro para				

		isolar o defeito.		
51	03.02	Não imprime e piscam luzes do painel	31/07/02	S
		Não imprime e piscam luzes do painel		
		Verificar no manual da impressora o significado das luzes. Normalmente é sinal de problema nos cartuchos ou cabeça de impressão.		
52	03.02	Puxa papel e não imprime. Pisca luz.	31/07/02	I
		Puxa papel e não imprime. Pisca luz do painel.		
		Provável problema no cartucho de impressão. Testar com outro. Se não for verificar conjunto bloco móvel de suporte aos cartuchos, pode estar quebrado.		
53	03.02	Barulho estranho ao imprimir	31/07/02	I
		Faz barulho estranho quando imprime		
		Fazer manutenção na impressora. Limpeza e lubrificação.		
54	03.02	Borra a impressão	31/07/02	I
		Borra a impressão		
		Fazer manutenção na impressora. Limpeza e lubrificação. Testar com outros cartuchos se não resolver.		
55	03.02	Só imprime caracteres estranhos	31/07/02	N
		Só imprime caracteres estranhos e não o que foi enviado		
		Problema na interface de comunicação da impressora ou do micro. Testar em outro micro para isolar a origem do defeito. Verificar se o driver usado é o correto.		
56	03.02	Não reconhece cartuchos	31/07/02	S
		Não reconhece cartuchos encaixados		
		Defeito no cartucho ou conjunto do bloco móvel com o encaixe dos cartuchos quebrado ou cabo flat ca cabeça de impressão quebrado ou oxidado.		
57	03.02	Imprime fora do alinhamento	31/07/02	I
		Imprime fora do alinhamento		
		Fazer alinhamento via software do driver da impressora. Caso não resolva deve ter folga entre cabeça de impressão e eixo. Trocar buchas.		
58	03.02	Não imprime colorido	31/07/02	S
		Não imprime colorido - nenhuma cor		
		Trocar cartucho color. Se não resolver verificar driver de impressão.		
59	03.02	Não imprime preto	31/07/02	S
		Não imprime preto		
		Trocar cartucho preto. Se não resolver verificar driver de impressão.		
60	03.02	Não imprime algumas cores	31/07/02	I
		Não imprime algumas cores		
		Trocar cartucho color. Se não resolver verificar driver de impressão.		
61	03.02	Imprime com linhas falhadas horizontais	31/07/02	I
		Imprime com linhas falhadas horizontais		
		Fazer limpeza da cabeça de impressão. Se não resolver trocar o cartucho ou a cabeça (no caso de cabeça de impressão fixa como nas Epson).		
62	03.02	Erro quando imprime muitas páginas	31/07/02	I
		Dá erro quando imprime muitas páginas		
		Pouco espaço no winchester do micro ou buffer da impressora muito		

		pequeno. Tentar mudar modo de spool no micro.		
63	03.03	Não liga	31/07/02	S
		Não liga		
		Problema de alimentação de energia. Trocar a fonte ou os componentes queimados da mesma.		
64	03.03	Não imprime e não dá sinal nenhum	31/07/02	S
		Não imprime e não dá nenhum sinal		
		Impressora não está recebendo dados. Provável defeito na interface de comunicação da impressora ou do micro. Testar em outro micro para isolar o defeito.		
65	03.03	Não imprime e piscam luzes do painel	31/07/02	S
		Não imprime e piscam luzes do painel		
		Verificar no manual da impressora o significado das luzes. Normalmente é sinal de papel atolado.		
66	03.03	Não puxa papel	31/07/02	I
		Não puxa papel		
		Verificar engrenagens do conjunto de tracionamento do papel.		
67	03.03	Barulho estranho ao imprimir	31/07/02	I
		Barulho estranho ao imprimir		
		Verificar engrenagens do conjunto de tracionamento do papel.		
68	03.03	Borra a impressão	31/07/02	I
		Borra a impressão		
		Conjunto de fusão não está funcionando e o toner está saindo em forma de pó. Verificar lâmpada de fusão.		
69	03.03	Só imprime caracteres estranhos	31/07/02	S
		Só imprime caracteres estranhos		
		Problema na interface de comunicação da impressora ou do micro. Testar em outro micro para isolar a origem do defeito. Verificar se o driver usado é o correto.		
70	03.03	Imprime com linhas falhadas verticais	31/07/02	S
		Imprime com linhas falhadas na vertical		
		Rolo de fusão riscado ou com marcas. Deve ser trocado.		
71	03.03	Atola papel	31/07/02	I
		Atola papel		
		Papel úmido ou engrenagens do conjunto tracionador gastas ou quebradas.		
72	04.01	Não segura quando falta luz	31/07/02	N
		Não mantém os equipamentos ligados quando falta luz		
		Bateria com defeito. Trocar a bateria. Pode ainda estar com pouca carga, após recarregar deve-se refazer o teste.		
73	04.01	Não liga	31/07/02	S
		Não liga		
		Problema de alimentação de energia. Trocar a fonte ou os componentes queimados da mesma.		
74	04.01	Mesmo com luz fica só na bateria	31/07/02	N
		Mesmo quando tem luz fica só na bateria (inversor)		
75	05.01	Não liga	31/07/02	S

	Não liga			
	Problema de alimentação de energia. Trocar a fonte ou os componentes queimados da mesma.			
76	05.01	Faz barulho de vibração	31/07/02	I
	Faz barulho de vibração quando está ligado			
	Transformador mal-fixado na carcaça ou carcaça mal fechada.			
77	05.01	Faz barulho de cliques	31/07/02	I
	Faz barulho de cliques quando está ligado			
	Relé de baixa qualidade, trocar. Pode ainda ser resultado de muita oscilação na tensão da rede elétrica.			
78	05.01	Cheiro de queimado	31/07/02	N
	Dá cheiro de queimado após um tempo ligado			
	Potência consumida acima da especificação do aparelho. Remover algum dos equipamentos do estabilizador.			
79	06.01	Não liga	31/07/02	S
	Não liga			
	Problema de alimentação de energia. Trocar a fonte ou os componentes queimados da mesma.			
80	06.01	Não comunica com o micro	31/07/02	S
	Liga mas não comunica com o micro			
	Reinstalar driver. Verificar se a porta paralela está em EPP na BIOS do micro.			
81	06.01	Não escaneia	31/07/02	S
	Não acusa erro mas não escaneia imagem nenhuma			
	Verificar se a lâmpada acende e se o carro do sensor se desloca.			
82	06.01	Não acende a lâmpada	31/07/02	S
	Não acende a lâmpada			
	Testar com outra lâmpada.			
83	06.01	Tranca durante o escaneamento	31/07/02	I
	O carro do scanner tranca antes do final do escaneamento			
	Fazer manutenção e lubrificar o eixo do carro do sensor/lâmpada. Verificar conexões internas.			

Anexo 4. O Código-Fonte do Algoritmo de Recuperação e Casamento – parte 1

```
unit NovoCaso;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  DBCtrls, StdCtrls, Buttons, ExtCtrls;

type
  TfrmNovoCaso01 = class(TForm)
    lbEquip: TListBox;
    Label1: TLabel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    Memo1: TMemo;
    Label2: TLabel;
    Bevel1: TBevel;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure lbEquipClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmNovoCaso01: TfrmNovoCaso01;
  i,j : integer;

implementation

uses Arquivos, NovoCaso02;

{$R *.DFM}

procedure TfrmNovoCaso01.SpeedButton1Click(Sender: TObject);
begin
  Close;
end;

procedure TfrmNovoCaso01.SpeedButton2Click(Sender: TObject);
begin
  With Dados.sqlConsultaPalavras Do
    Begin
```

```

Close;
sql.Clear;
sql.Add('Select * From Dicionar Where');
j := 0;
For i := 1 to frmNovoCaso01.lbEquip.Items.Count Do
Begin
  If frmNovoCaso01.lbEquip.Selected[i-1] = True Then
  Begin
    if j = 0 Then
    Begin
      sql.Add('Cod_Estrut = ' + Copy(lbEquip.Items.Strings[i-1],1,5));
      j := 1;
    End
  Else
    sql.Add('OR Cod_Estrut = ' + Copy(lbEquip.Items.Strings[i-1],1,5));
  End;
End;
sql.Add('Order By Palavra');
Open;
End;
Close;
frmNovoCaso02.Show;
end;

```

```

procedure TfrmNovoCaso01.lbEquipClick(Sender: TObject);
Var
  Inicio:String;
begin
  If Copy(lbEquip.Items.Strings[lbEquip.ItemIndex],4,2) = '00' Then
  Begin
    Inicio := Copy(lbEquip.Items.Strings[lbEquip.ItemIndex],1,2);
    For i := lbEquip.ItemIndex + 2 to frmNovoCaso01.lbEquip.Items.Count Do
    Begin
      If Copy(lbEquip.Items.Strings[i-1],1,2) = Inicio Then
        frmNovoCaso01.lbEquip.Selected[i-1] := Not
        frmNovoCaso01.lbEquip.Selected[i-1];
      End;
    End;
  End;
end;

```

```

procedure TfrmNovoCaso01.FormCreate(Sender: TObject);
begin
  Dados.TabArvore.First;
  lbEquip.Items.Clear;
  While Not Dados.TabArvore.Eof Do
  Begin

```

```
        lbEquip.Items.Add(Dados.TabArvoreCOD_ESTRUT.Value + ' - ' +  
        Dados.TabArvoreDESCRICAO.Value);  
        Dados.TabArvore.Next;  
    End;  
end;  
  
end.
```

Anexo 5. O Código-Fonte do Algoritmo de Recuperação e Casamento – parte 2

```
unit NovoCaso02;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls, Grids, DBGrids, Buttons, ExtCtrls;
```

```
type
```

```
TfrmNovoCaso02 = class(TForm)  
  lbPalavras: TListBox;  
  Label1: TLabel;  
  Label2: TLabel;  
  Memo1: TMemo;  
  SpeedButton2: TSpeedButton;  
  SpeedButton1: TSpeedButton;  
  SpeedButton3: TSpeedButton;  
  Bevel1: TBevel;  
  procedure FormShow(Sender: TObject);  
  procedure SpeedButton1Click(Sender: TObject);  
  procedure SpeedButton2Click(Sender: TObject);  
  procedure Adiciona(Caso:integer;Grau:integer);  
  procedure Classificar;  
  procedure SpeedButton3Click(Sender: TObject);  
private  
  { Private declarations }  
public  
  R:array[1..100,1..2] Of Integer;  
  Ultimo:Integer;  
  { Public declarations }  
end;
```

```
var
```

```
frmNovoCaso02: TfrmNovoCaso02;  
I,J:Integer;
```

```
implementation
```

```
uses Arquivos, NovoCaso, NovoCaso03;
```

```
{ $R *.DFM }
```

```
procedure TfrmNovoCaso02.FormShow(Sender: TObject);
```

```
Var
```

```
Palavra : String;
```

```
begin
```

```

lbPalavras.Items.Clear;
Ultimo := 0;
For i:= 0 To 100 Do
Begin
    R[I,1] := 0;
    R[I,2] := 0;
End;

Palavra:="";
Dados.sqlConsultaPalavras.First;
While Not Dados.sqlConsultaPalavras.Eof Do
Begin
    If Palavra <> Dados.sqlConsultaPalavrasPALAVRA.Value Then
    Begin
        lbPalavras.Items.Add(Dados.sqlConsultaPalavrasPALAVRA.Value);
        Palavra := Dados.sqlConsultaPalavrasPALAVRA.Value;
    End;
    Dados.sqlConsultaPalavras.Next;
End;
end;

procedure TfrmNovoCaso02.SpeedButton1Click(Sender: TObject);
begin
    Close;
end;

procedure TfrmNovoCaso02.Adiciona(Caso:integer; Grau:integer);
Var
    Achou:Boolean;
Begin
    Achou := False;
    For I:= 0 to Ultimo Do
        If R[I, 1] = Caso Then
        Begin
            R[I,2] := R[I,2] + Grau;
            Achou := True;
        End;
    If Achou = False Then
    Begin
        R[Ultimo,1] := Caso;
        R[Ultimo,2] := Grau;
        Ultimo := Ultimo + 1;
    End;
End;

procedure TfrmNovoCaso02.Classificar;
Var
    AuxCaso, AuxGrau:Integer;

```

```

Begin
  //Método de Ordenação BOLHA
  For I:= 0 to Ultimo-1 Do
    For J:= I+1 to Ultimo Do
      If R[I, 2] < R[J,2] Then
        Begin
          AuxCaso := R[I,1];
          AuxGrau := R[I,2];
          R[I,1] := R[J,1];
          R[I,2] := R[J,2];
          R[J,1] := AuxCaso;
          R[J,2] := AuxGrau;
        End;
      End;
    End;
  End;

procedure TfrmNovoCaso02.SpeedButton2Click(Sender: TObject);
begin
  Dados.sqlConsultaPalavras.First;
  While Not Dados.sqlConsultaPalavras.EOF Do
    Begin
      If
lbPalavras.Selected[lbPalavras.Items.IndexOf(Dados.sqlConsultaPalavrasPALAVRA.V
alue)] = True Then

        Adiciona(StrToInt(FloatToStr(Dados.sqlConsultaPalavrasCOD_CASO.Value)),
StrToInt(FloatToStr(Dados.sqlConsultaPalavrasGRAU.Value)));
        Dados.sqlConsultaPalavras.Next;
      End;
      Classificar;
      Close;
      frmNovoCaso03.Show;
    end;

procedure TfrmNovoCaso02.SpeedButton3Click(Sender: TObject);
begin
  Close;
  frmNovoCaso01.Show;
end;

end.

```

Anexo 6. O Código-Fonte do Algoritmo de Recuperação e Casamento – parte 3

```
unit NovoCaso03;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
Buttons, StdCtrls, Mask, DBCtrls, ExtCtrls;
```

```
type
```

```
TfrmNovoCaso03 = class(TForm)  
    Label2: TLabel;  
    Label1: TLabel;  
    DBEdit1: TDBEdit;  
    Label3: TLabel;  
    DBEdit2: TDBEdit;  
    Label4: TLabel;  
    DBEdit3: TDBEdit;  
    Label5: TLabel;  
    DBEdit4: TDBEdit;  
    Label6: TLabel;  
    DBEdit5: TDBEdit;  
    Label7: TLabel;  
    DBEdit6: TDBEdit;  
    btnRetornarTela: TSpeedButton;  
    btnAvancar: TSpeedButton;  
    SpeedButton1: TSpeedButton;  
    btnRetornar: TSpeedButton;  
    lblGrau: TStaticText;  
    Label9: TLabel;  
    SpeedButton2: TSpeedButton;  
    Bevel1: TBevel;  
    DBRadioGroup1: TDBRadioGroup;  
    SpeedButton3: TSpeedButton;  
    procedure SpeedButton1Click(Sender: TObject);  
    procedure SpeedButton4Click(Sender: TObject);  
    procedure btnRetornarTelaClick(Sender: TObject);  
    procedure FormShow(Sender: TObject);  
    procedure btnAvancarClick(Sender: TObject);  
    procedure btnRetornarClick(Sender: TObject);  
    procedure SpeedButton2Click(Sender: TObject);  
    procedure SpeedButton3Click(Sender: TObject);  
private  
    { Private declarations }  
public  
    { Public declarations }  
end;
```



```

var
  frmNovoCaso03: TfrmNovoCaso03;
  Caso,I,J:Integer;
implementation

uses Arquivos, NovoCaso02, RCaso, RCasosPossiveis;

{$R *.DFM}

procedure TfrmNovoCaso03.SpeedButton1Click(Sender: TObject);
begin
  Close;
end;

procedure TfrmNovoCaso03.SpeedButton4Click(Sender: TObject);
begin
  Close;
  frmNovoCaso02.Show;
end;

procedure TfrmNovoCaso03.btnRetornarTelaClick(Sender: TObject);
begin
  Close;
  frmNovoCaso02.Show;
end;

procedure TfrmNovoCaso03.FormShow(Sender: TObject);
begin
  Caso := 0;
  Dados.TabCasos.Locate('Cod_Caso', frmNovoCaso02.R[Caso,1] , []);
  If frmNovoCaso02.Ultimo > 1 Then
    btnAvancar.Enabled := True
  Else
    btnAvancar.Enabled := False;
    btnRetornar.Enabled := False;
    lblGrau.Caption := IntToStr(frmNovoCaso02.R[Caso,2]);
end;

procedure TfrmNovoCaso03.btnAvancarClick(Sender: TObject);
begin
  Caso := Caso + 1;
  If Dados.TabCasos.Locate('Cod_Caso', frmNovoCaso02.R[Caso,1] , []) Then
    Begin
      If frmNovoCaso02.Ultimo <> Caso+1 Then
        btnAvancar.Enabled := True
      Else
        btnAvancar.Enabled := False;
    End
  End
end;

```

```

        btnRetornar.Enabled := True;
        lblGrau.Caption := IntToStr(frmNovoCaso02.R[Caso,2]);
    End
    Else
        Caso := Caso - 1;
end;

procedure TfrmNovoCaso03.btnRetornarClick(Sender: TObject);
begin
    Caso := Caso - 1;
    Dados.TabCasos.Locate('Cod_Caso', frmNovoCaso02.R[Caso,1] , []);
    If Caso > 0 Then
        btnRetornar.Enabled := True
    Else
        btnRetornar.Enabled := False;
        btnAvancar.Enabled := True;
        lblGrau.Caption := IntToStr(frmNovoCaso02.R[Caso,2]);
end;

procedure TfrmNovoCaso03.SpeedButton2Click(Sender: TObject);
begin
    relCaso.QuickRep1.Preview; //Trocar .Preview por .Print para impressão direta
end;

procedure TfrmNovoCaso03.SpeedButton3Click(Sender: TObject);
begin
    With Dados.sqlCasosPossiveis Do
    Begin
        Close;
        sql.Clear;
        sql.Add('Select * from Casos Where');
        j := 0;
        For i := 0 to frmNovoCaso02.Ultimo-1 Do
        Begin
            if j = 0 Then
            Begin
                sql.Add('(Cod_Caso = ' + IntToStr(frmNovoCaso02.R[i,1])+')');
                j := 1;
            End
            Else
                sql.Add('Or (Cod_Caso = ' + IntToStr(frmNovoCaso02.R[i,1])+')');
            End;
        End;
        Open;
    End;
    relCasosPossiveis.QuickRep1.Preview;
    Dados.sqlCasosPossiveis.Close;
end;
end.

```

5. Bibliografia

- AAMODT, A; PLAZA, E. *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*. AI Communications, v.7, n.1, p.39-59, Mar. 1994.
- ABEL, Mara. *Um Estudo sobre Raciocínio Baseado em Casos*. Trabalho Individual. Porto Alegre: CPGCC da UFRGS, 1996.
- ADELI, Hojjat, et al. *Knowledge Engineering: Vol. II – Applications*. The Ohio State University. New York: McGraw Hill, 1990.
- BARRETO, Jorge M. *Inteligência Artificial: no limiar do século XXI*. Florianópolis: Duplic, 1999.
- BRIDGE, Derek. *Product Recommendation Systems: A New Direction*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... Cork, 2001. 8 p.
- CHORAFAS, Dimitris N. *Sistemas especialistas: aplicações comerciais*. São Paulo: Ed. McGraw Hill, 1988
- CUNNINGHAM, Pádraig, et al. *Intelligent Support for Online Sales: The WEBSELL Experience*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... Dublin, 2001. 6 p.
- FERREIRA, Aurélio Buarque de Holanda. *Dicionário Aurélio da Língua Portuguesa*. São Paulo: Ed. Nova Fronteira, 2001.
- FOX, Susan. Introspective Learning For Case-Base Planning. 1995. 270 f. Tese (Doctor of Philosophy in the Department of Computer Science), Indiana University, Indiana.
- JEDLITSCHKA, Andreas, et al. *Corporate Information Network (COIN): The Fraunhofer IESE Experience Factory*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... Kaiserlautern: IESE, 2001. 4 p.
- HANSEN, Bjarne K; RIORDAN, Denis. *Weather Prediction Using Case-Based Reasoning and Fuzzy Set Theory*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... New Brunswick, 2001. 2 p.
- HARMON, Paul; KING, David. *Expert Systems: Artificial Intelligence in Business*. New York: Wiley Press Book, 1985.

- HAYES-ROTH, Frederick. *Building Expert Systems*. Stanford: Addison-Wesley Publishing Company, Inc, 1983.
- KOLODNER, Janet. *Case-Based Reasoning*. San Mateo: Morgan Kaufmann, 1993.
- LEWIS, Lundy. *A Case-Based Reasoning Approach to the Management of faults in Communications Networks*. IEEE INFOCOM, San Francisco, v.3, p.1422-1429, Apr. 1993. Trabalho apresentado no Annual Joint Conference of The IEEE Computer and Communications Societies, 20., 1993.
- LIEBOWITZ, Jay. *Introduction to expert systems*. Santa Cruz, Califórnia: Mitchell Publishing, 1988.
- MALEK, Maria; KANAWATI, Rushed. *A Cooperating Hybrid Neural-CBR Classifiers for Building On-line*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... Cergy, 2000. 5 p.
- McSHERRY, David. *Improving the Build Quality of Case-Based Reasoning Systems: the Case-Authoring Challenge*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... Coleraine: Ulster, 2001. 6 p.
- MELCHIORIS, Cristina. *Raciocínio Baseado em Casos Aplicado ao Gerenciamento de Falhas em Redes de Computadores*. 1999. 151 f. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- MÜLLER, Isabela R. F. *Uma análise das aplicabilidades de sistemas especialistas na área de gestão: um estudo de caso*. 1998. 131 f. Dissertação (Mestrado em Administração) – Curso de Pós-graduação em Administração, Universidade Federal de Santa Catarina, Florianópolis.
- MULLINS, Mark; SMYTH, Barry. *Visualisation Methods in Case-Based Reasoning*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... Dublin, 2001. 4 p.
- NEWEL, Allan e Simon, Herbert. *A Program that simulates human thought*. New York: Academic Press, 1961.
- PARTRIDGE, D. e Hussain, K. *Knowledge-based information systems*. Londres: McGraw Hill, 1995.
- RIBEIRO, Horácio da Cunha e Sousa. *Introdução aos Sistemas Especialistas*. São Paulo: Livros Técnicos e Científicos Editora S.A., 1987.
- RIO, Beatriz Z., FERNÁNDEZ, Rodolfo C. *Sistemas Expertos: conceptos generales y su aplicacion comercial*. Tecnologia em Marcha. V.9, n.4, 1990.

- ROTH-BERGHOFER, Thomas. *Searching Online Advertisements with Textual Case-Based Reasoning*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... Kaiserslautern, 2001. 2 p.
- SCHANK, Roger C. *Dynamic memory: a theory of learning in computers and people*. Cambridge: University Press, 1982.
- SCHANK, Roger C. *The Cognitive Computer: on language learning and artificial intelligence*. Pennsauken: Addison-Wesley Publishing Company, Inc, 1984.
- SCHILDT, Herbert. *Advanced turbo prolog*. Berkley: McGraw Hill, 1987.
- SOVAT, Ricardo B.; CARVALHO, André C. P. L. F. *Retrieval and Adaptation os Cases Using an Artificial Neural Network*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... São Carlos: USP, 2001. 5 p.
- TEIXEIRA, Alisson R. *A utilização de programas de computador como agentes no processo de transferência da informação: criação e avaliação de um sistema especialista baseado em casos*. 2000. 92 f. Dissertação (Mestrado em Ciência da Informação) – Curso de Pós-graduação em Ciência da Informação, Universidade de Brasília, Brasília.
- YAGER, Ronald R. *Induced OWA Aggregation in Case Based Reasoning*. In: International Conference on Case-Based Reasoning, 4., 2001, Vancouver. Proceedings... New Rochelle, 2001. 5 p.
- YAZDANI, Masoud. *Artificial Intelligence: principles and applications*. Chapman and Hall Ltd. London. 1986.
- WATSON, Ian; MARIR, Farhi. *Case-based Reasoning: A Review*. The Knowledge Engineering Review. London. V.9. n.4. 1994.
- WEISS, Sholom M. e Kulikowski. *Guia prático para projetar sistemas especialistas*. Rio de Janeiro: LTC, 1988.